# A Software-based Eye Tracking System for the Study of Air-traffic Displays

Jeffrey B. Mulligan
NASA Ames Research Center
MS 262-2
Moffett Field, CA, 94035-1000
(650) 604-3745
jmulligan@mail.arc.nasa.gov

## ABSTRACT

This paper describes a software-based system for offline tracking of eye and head movements using stored video images, designed for use in the study of air-traffic displays. These displays are typically dense with information; to address the research questions, we wish to be able to localize gaze within a single word within a line of text (a few minutes of arc), while at the same time allowing some freedom of movement to the subject. Accurate gaze tracking in the presence of head movements requires high precision head tracking, and this was accomplished by registration of images from a forward-looking scene camera with a narrow field of view.

## Keywords

Head and eye tracking, image registration, scan-path analysis, air traffic displays.

## 1. INTRODUCTION

The measurement of eye movements in free-head conditions during the performance of natural tasks is a potentially useful tool in human factors research. Although magnetic trackers can be used to track head movement, they require instrumentation of the environment, and may not function well in the presence of other equipment (e.g. the magnetic environment of a flight simulator cab). For these reasons, we have developed a system for free-head eye movement measurement in which images from a forward-looking scene camera are used to recover head position, in order to refer head-relative gaze measurements back to the environment. This paper describes the procedures which implement this system.

## 2. IMAGING COLLECTION SYSTEM

### 2.1 Head-mounted Camera System

Our head-mounted camera system is based on a commercial unit sold by ISCAN. There are two cameras: an infrared eye camera and a visible light forward-looking scene camera. The signals from these cameras are connected to the R and G inputs of an RGB frame grabber (Matrox Meteor 1). This approach to multi-channel capture requires that the two cameras be synchronized (genlocked). Because the eye camera provided by ISCAN does not accept external sync, its signal is used as the master sync signal for the entire system. Unfortunately, this signal is not RS170A compliant, e.g. the horizontal and vertical frequencies are not derived from the color subcarrier frequency. For this reason, it was impossible to synchronize the color scene camera provided by ISCAN, which was replaced by a similar monochrome camera (ELMO model # ME-411E). The original scene camera had a rather wide field of view; to increase accuracy of head-tracking, the replacement camera was fitted with a 15 mm focal length lens, providing a narrower field of view of approximately 15 degrees.

In the ISCAN design, a two inch "hot mirror" is suspended in front of the left eye at a 45 degree angle. This mirror is transparent to visible light, and the subject looks through it to view the environment. The LED illuminator and eye camera are situated above this mirror and use it to illuminate and view the eye. The scene camera is a small "lipstick" camera mounted horizontally below the left eye. In the original ISCAN design, its line of sight was reflected twice: first by a small silver mirror mounted on a small stalk, which redirected the line of sight upward; and secondly by the hot mirror, which redirected the line of sight in the forward direction, more-or-less coincident with the left eye's line of sight.

A problem with this design was revealed when the focal length of the scene camera lens was increased: Because the hot mirror has a thickness of approximately 3 millimeters, there are separate reflections from each of the two surfaces. With the original scene camera lens, the positions of these two images differed by less than 1 pixel, but with the longer lens an annoying double image was seen. The position of the small mirror stalk was therefore changed to redirect the scene camera line of sight forward instead of upward, taking the hot mirror out of the optical path. This solved the double image problem, at the expense of lowering the scene camera line of sight approximately an inch below the line of sight of the subject's eye [1].

[1] An additional minor complication introduced by this change was left-right reversal of the image, but this is easily corrected in post-processing.

## 2.2 Timing Reference

In addition to providing a genlock input for the scene camera, the eye camera signal is also routed to a master sync generator, which provides "house sync" to the frame grabber and other equipment in the laboratory. In order to monitor the integrity of the recording system, the third input channel was used to record a "clock" signal from a fixed camera located in another room. This camera viewed a four digit LED display driven by a counter clocked by the vertical drive sync signal. When the recording system is functioning properly, the counts in the initial and final fields differ by the number of recorded fields (less one). If a frame is "dropped," however, due to the inability of the disk subsystem to keep up, a count will be missing at the corresponding point in the recorded sequence. corrected in post-processing. In the following subsections, we present each of these steps in more detail.

## 2.3 Image Capture

The three video signals are digitized using a Matrox Meteor 1 frame grabber board, hosted by a dual-Pentium PC running the Linux operating system. Images are stored in real-time to an array of 8 external disk drives (IBM, ultra SCSI, 10000 rpm). Although these drives are quite fast (approx. 28 Mb/sec), a single drive is still too slow to handle the full-resolution data rate (37 Mb/sec). We therefore use multiple drives, both to increase the data rate through parallelism, and to increase the total recording capacity. To minimize the software overhead, and to insure sequential access to the drives, they are not formatted as a normal Linux file system, rather the application program writes directly to the device files. The pthreads library is used to create one thread of execution per disk; this effectively provides an asynchronous write capability which allows multiple drives to be written to simultaneously.

## 2.4 Subsampling and Storage

There are a number of problems with the raw data stored to the disk array by the acquisition program. First, because the disks are not formatted as standard file systems, they cannot be exported to other machines on the network, but must be read by an application program on the local machine. Secondly, the frame grabber uses one 32 long word of memory to store 24 bits of actual data from the cameras; of these 24 bits, we really only care about 16 (corresponding to the two head-mounted cameras). We inspect the temporal reference for correctness, but do not store it. Thirdly, the images are stored as interlaced video frames. Because the two fields that make up a single frame were acquired at different times, we deinterlace the frames and analyse the sequences field-by-field. But because of interlacing, the vertical resolution of the fields is less than half of the horizontal resolution. In order to work with images which have "square pixels" we need to subsample in the horizontal dimension by a factor of two. Lastly, the full resolution images comprise an enormous amount of data, both to store and to process. We therefore subsample by an additional factor of 2 in both the horizontal and vertical dimensions.

The method used for subsampling is described by Watson [4]. In brief, the images are cropped in the horizontal dimension to 512 columns, and zero-padded in the vertical dimension to 256 rows. The Fourier transform is then computed; a subregion of the transform is extracted and inverse-transformed into the reduced-resolution target image. This is done at several resolutions: 256x256, 128x128, 64x64 and 32x32. To date, we have not used the smaller resolutions, but they require relatively little extra computation and storage space, and will allow us to study how resolution and accuracy trade off for various algorithms. Finally, the subsampled images are stored as JPEG files. These files are stored on a large RAID filesystem which is shared among a cluster of workstations, and they are also copied to CD ROM for archival storage.

# 3. HEAD TRACKING METHODS

We are not interested in head-tracking per se, but in order to refer measurements from a head-mounted eye camera back to objects in the environment, it is necessary to know the angular relationship of these objects to the head. Our approach is to match images of the stimulus display captured by the head-mounted scene camera to synthetic models of the display content. Normalized cross-correlation is the primary method used to perform the matching. Because we do not model elements of the environment outside of the display screen, we first segment the display area from the monitor bezel and surrounding objects, which are masked so as to be excluded from succeeding computations.

Small head rotations in the yaw and pitch directions, as well as small side-to-side translations, are manifested as horizontal and vertical displacements of the scene camera image. These translations can be recovered from a single cross-correlation, provided the images are matched for rotation (roll) and scale (zoom). In an ideal situation, a small pitch of the head would produce a purely vertical translation of the scene camera image. In general, this is not the case, however, because of minor misalignment of the camera and mirror in the head mount and variability in the fitting of the head mount on the subject's head. These factors are constant within a run, but there are also dynamic changes in roll posture that must be tracked within a trial. Similarly, the primary determinants of scale factor relating the scene camera image and the display model are fixed parameters (lens focal length and distance to the display). In addition to these fixed parameters, however, postural adjustments that change the distance between the head and the display will produce corresponding perturbations in the parameter values.

Determination of roll and zoom parameters is done by searching the two dimensional parameter space for the pair of values which produces the highest numerical value of the normalized cross correlation. Because this iterative search has a high computational cost, we avoid performing it on each frame. Instead, we perform it on the initial frame of each segment in a preprocessing step. These initial values are used for subsequent frames, and the correlation value is tracked and used to determine when the parameters should be re-estimated.

In the following subsections, we present each of these steps in more detail.

## 3.1 Region Selection

The first step is to segment the display area from the monitor bezel and surrounding clutter. A typical image is shown in figure 1. We take advantage of the fact that the display consists mainly of a dark background, with a few light pixels. We can therefore do a simple gray-level selection which will include the display area, as well as any surrounding areas having the same dark luminance. We then apply a number of morphological operators [2]. We first dilate the image in order to fill in "holes" in the display area produced by the aircraft icons. We then apply a complementary erosion to restore the boundary to its original position. Additional erosions eliminate "bridges" crossing the monitor bezel connecting the display area to dark regions outside; the area around the manufacturer's logo proved particularly troublesome in this regard.

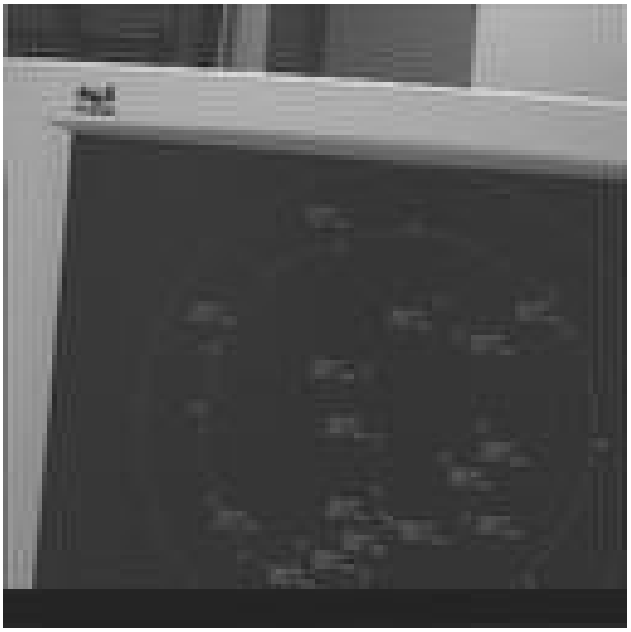

**Figure 1. Raw scene camera image after subsampling to 128x128, and JPEG encoding. The scene camera has captured the majority of the screen content, as well as the upper left hand portion of the monitor bezel, and a bit of the background.**

At this point, we have represented the image as a binary image consisting of one or more disconnected components. (At least, we hope that the component corresponding to the display area is disconnected from all other components!) We proceed by searching for the largest component. We isolate individual components as follows: we choose a seed point at the first non-zero pixel, and perform a flood-fill with a large value. We identify the filled region by thresholding, compute its area, and store. We then subtract this component from our working image, and repeat the process until there are no non-zero pixels in the working image.

After the largest component has been identified in this manner, we apply dilation to restore the boundaries to their original position. Provided that the head-mount was properly aligned on the subject's head at the time of the experiment, the display area is the largest region in the scene camera image, and this method works well.

## 3.2 Model Construction

The display monitor is a 15" LCD flat panel display (Viewsonic VG150). We chose a flat panel instead of a CRT display because the continuously illuminated LCD display does not produce a beat pattern with the scan frequency of the scene camera. The stimuli fill the entire display area, which has a resolution of 768 rows by 1024 columns. The program which presents the stimuli draws using specifications from precomputed configuration files. The stimulus descriptions can also be used to recreate the stimuli in a window which can be read back and saved to disk. This is done for each stimulus, and then a second pass is performed in which the luminance is computed as a weighted sum of the R,G and B values. (At this point we have not attempted to measure or correct for the monitor gamma, or the exact RGB luminance ratios, although it should be done.)

## 3.3 Pre-filtering

We wish to match the aircraft icons and data tags between the scene camera image and the display model, without regard to absolute pixel values. We therefore apply a band-pass filter to both the input and the model. The choice of passband is determined by trial and error. For most of our work, we have used the 128x128 versions of the scene camera image, embedded in a 256x256 staging area. The display model is warped to its own 256x256 staging area using a two-parameter affine transformation consisting of a rotation and zoom about the center of the image. Both images are filtered with a difference-of-Gaussians bandpass filter. The main benefit from this comes from zeroing of the mean level and slow illumination variations.

## 3.4 Normalized Cross-correlation

The best translational alignment between the filtered scene camera image and the model is obtained using normalized cross-correlation. Sub-pixel localization of the maximum is performed as has been described previously [3]. In brief, we find the pixel in the correlation image with the maximum value, and then fit the 3x3 neighborhood centered on this pixel with a quadratic surface. From the coefficients describing this surface we can compute both the location and the value of the interpolated peak.

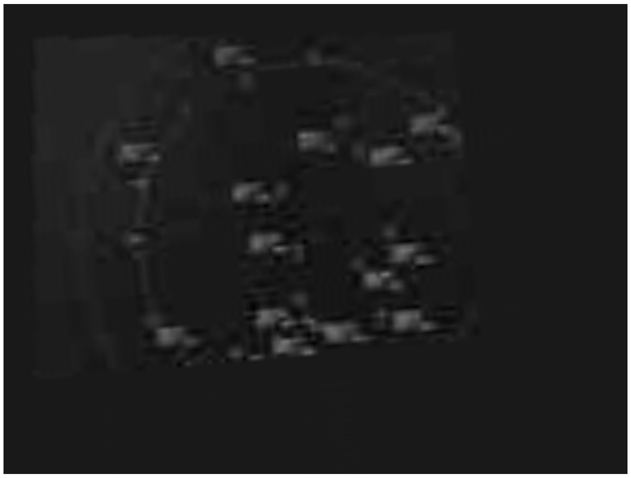

**Figure 2. The image from figure 1, after extraction of the scene area, and expansion and rotation to register with the original stimulus, at a resolution of 768x1024.**

-

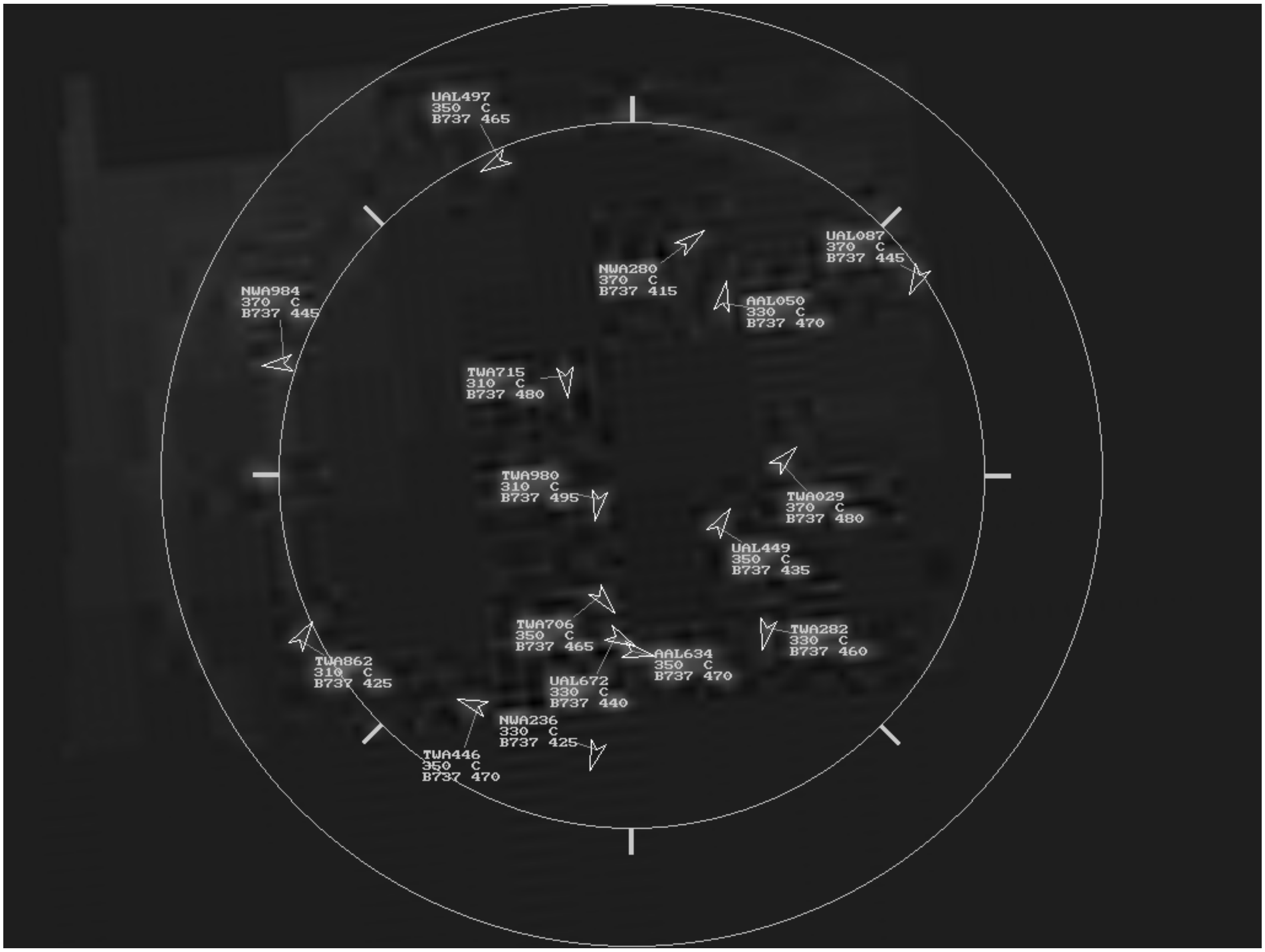

**Figure 3. The image from figure 2, with the original stimulus redrawn over it to allow visualization of the quality of registration.**

## 3.5 Zoom and Roll Parameter Search

The cross-correlations are computed relative to the (subsampled) scene camera image. Because the resolution of the display model is much larger (768x1024) it must be down-sampled to a size and orientation that matches that of the scene camera image. We accomplish this using a general purpose warping routine which takes as an input a complex-valued map control image having the same resolution as the source image. Each pixel in the map image contains the coordinates in the destination image to which the corresponding source pixel should be mapped. Mapped pixel values are added into the destination image. If the target location falls between grid locations, the pixel value is split between the nearby locations using a linear distance-based weighting.

The approximate range of the zoom and roll parameters has been determined by trial-and-error. We use the STEPIT optimization routine [1] to search for an optimal setting. When the initial values are near the true values, STEPIT usually finds the correct answer. When the starting values are sufficiently off, however, STEPIT can become trapped in a local maximum. This condition can be detected by an unusually low value of the final correlation, in which case a coarse grid search is performed over the starting values; e.g., if our first attempt with STEPIT fails, we then perform multiple attempts for an array of starting values and tabulate the result. With our grid increment of 0.25 for scale factor (with solutions typically in the range 5-7) and 10 degrees for roll angle, we have been able to find the correct solution in every case, but this procedure is very time consuming.

## 3.6 Zoom and Roll Updating

The procedure described in the previous section is performed on all segments to determine the zoom and roll settings at the beginning of a trial. Because the subjects generally sit still, these values do not change a lot during a trial, and we do not need to recompute them for every frame. Small postural adjustments can cause them to vary, however, so we need to monitor the value of the correlation and occasionally reestimate the roll and zoom parameters when the correlation value falls below a threshold.

## 3.7 Comparison to Magnetic Tracking

For use in the laboratory, we have equipped the head mount with a magnetic tracking device (Polhemus InsideTrak). This system consists of an ISA circuit card installed in a PC, a transmitter whose position is fixed near the subject's head, and a small receiver mounted on the head mount. Typical data are shown in figures 4-6. This set of figures shows data from a brief recording during which the subject turned his head from side to side. The primary effect of the motion is seen as a roughly sinusoidal variation in horizontal position and yaw angle in the tracker recordings, and in horizontal registration offset in the scene camera data. The scene camera data in figure 6 are inverted with respect to the data in figures 4 and 5, and the traces are advanced by approximately 100 milliseconds due to an uncorrected system clock ansynchrony between the two machines used to make the recordings.

The main point of these figures is to show that the data obtained from the scene camera images are no more noisy than those obtained using the magnetic tracking device. Furthermore, they provide a direct measure of the positions of scene objects with respect to the visual directions measured by analysis of the eye images. We have not yet developed software to use the magnetic tracker data for gaze tracking, but the path is clear: first the spatial relations between the receiver and the eye must be determined, so that the motion of the eye in space can be computed from the motion of the receiver. (These will in general be different whenever rotations are involved.) Secondly, the tracker coordinate system (centered at the transmitter) must be referenced to the display; ideally, the spatial relation between the display and the transmitter would be fixed with some sort of rigid jig. When these things are done, the tracking data can be used to compute the position and orientation of the eye relative to the display, and to refer head-relative gaze measures to the display. It should be apparent, however, that because all 6 measures from the tracker must enter into these calculations, the effects of noise in the individual measurements will be cumulative. The image-based measure, while not recovering all six degrees of freedom, does provide the information needed to identify gaze targets, and does so in a much more direct way. We expect that its performance will exceed that possible with the magnetic tracking device, while being more portable and immune to hostile magnetic environments.

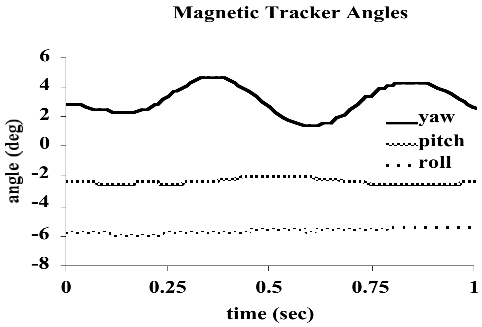

**Figure 4. Angular data from a magnetic sensor attached to the headmount while the subject turned his head from side to side.**

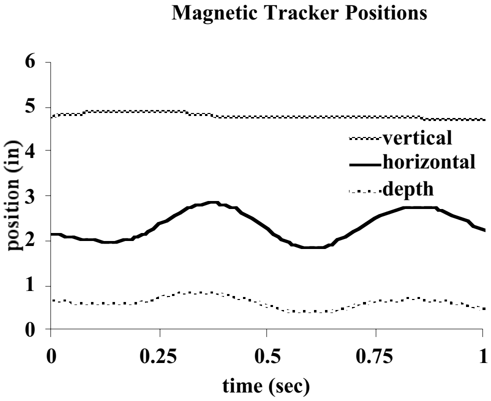

**Figure 5. Position data from a magnetic sensor attached to the headmount while the subject turned his head from side to side.**

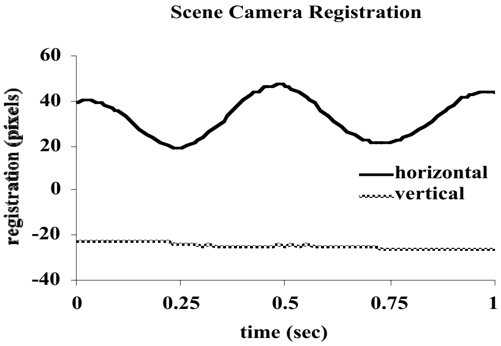

**Figure 6. Image registration data computed from the scene camera images collected during the movement shown in figures 4 and 5. The horizontal data are inverted with respect to the data in figures 4 and 5, and the traces are advanced by approximately 100 msec due to an uncorrected clock offset between the machines used to make the recordings. These data exhibit a noise level less than or equal to that in the magnetic tracker, and can be directly used for referring gaze measures to scene content.**

# 4. ANALYSIS OF EYE IMAGES

Images of the eye from the head mounted camera are processed to produce measurements of the locations of the pupil and the corneal reflex (CR). The issues are not particular to the free-head environment discussed here; nevertheless they will be presented in some detail, because the present methods have evolved significantly from previously described implementations [3].

Because the eye camera is fixed relative to the head, either one of these is sufficient in principle to determine head-relative gaze. Collection of both measures, however, may allow detection of head-mount slippage, and perhaps compensation. Some systems use the difference of pupil and CR measures as a translation-invariant measure of eye rotation. This is strictly correct only when the illuminator is at optical infinity; for nearer distances, the optimal combination weights (determined using a ray-tracing simulation) are of opposite signs but unequal in magnitude.

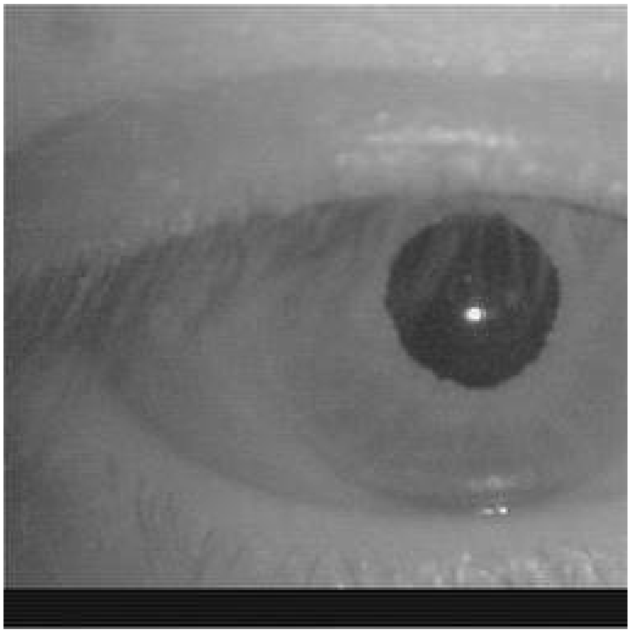

**Figure 7. A raw eye image from the head-mounted eye camera, after subsampling and JPEG encoding. The large, bright highlight near the center of the pupil is the corneal reflex, or first Purkinje image. Also visible are the third Purkinje image (the blurred spot to the right and slightly above the CR), and the fourth Purkinje image (the small spot above and slightly to the left of the CR).**

The illumination provided by the ISCAN head mount is very uniform, and there are few problems with shadows and other illumination-related problems. Even so, a simple threshold-based approach to finding the pupil fails for some subjects (mascara is especially problematic). We have therefore adopted a multistage approach: we first use thresholding to create a "pupil mask," a conservative estimate of a region containing the pupil. For a higher-precision measurement, an ellipse is fit to the pupil margin. This has the advantage of robustness to partial occlusion of the pupil by the upper eyelid (as in the vicinity of a blink).

## 4.1 Determination of Level Parameters

The initial processing steps use parameters representing the range of pixel values encountered in the pupil and iris. In early implementations of the system, these values were determined by having an operator indicate with the cursor regions within the pupil and iris for a few representative images. These values tend to be stable from run to run for a given subject, so this process did not generally have to be done for each run.

The current version of the system uses an automated procedure based on histogram analysis. This is now performed for each trial. Each movie is scanned, and the cumulative gray level histrogram is computed. Next, the peaks and troughs in the histrogram are identified, along with their respective counts. The first significant peak corresponds to the pixels in the pupil region, and we assume that the corresponding gray level is the pupil mean value. We assign the pupil maximum to be the value corresponding to the next trough, and subtract it from twice the pupil mean value to obtain our estimate of the pupil minimum value. The bulk of the gray level distribution comes from pixels in the iris, sclera, and eyelid skin. We are primarily interested in discriminating these areas from specular highlights such as the corneal reflex, so we adopt a conservative criterion. We take the iris min value to be 8 levels above the pupil max value, and take the location of the second highest peak as the iris max value. The software allows the user to tweak which histogram features are used for each setting, to allow adjustment to different illumination conditions found in other set-ups.

## 4.2 Highlight Processing

The first step is the construction of a pupil mask, i.e. a binary image representing the area of the pupil. Because we use gray level information to do this, it is convenient to first remove specular highlights from the image. The most obvious specular highlight is the corneal reflex, which often falls within the pupil, but it is also common to encounter smaller highlights on parts of the sclera and at the margins of the eyelids where tears may collect. We do this by detecting all regions whose value falls above the iris_max parameter. Pixels in these regions are clipped to a value of iris_max.

## 4.3 Pupil Mask Construction

After highlights have been removed as described above, it is fairly straightforward to find the pupil region using threshold-based methods. A binary mask is constructed by thresholding at the pupil_max value (fig. 8a); this image is first dilated to fill in "holes" within the pupil area, and then eroded to eliminate "islands" outside the pupil (fig. 8b). A second mask is also constructed, which we will use later to find the corneal reflex, which we call cr_mask. This is done by additional dilation of the pupil mask.

The area and position of the pupil mask are computed, as well as the "compactness," which is defined as the area divided by the square of the perimeter. These values are tested against limits, which if exceeded indicate a blink, or some other failure to correctly process the image.

The location of the pupil mask can be used to restrict subsequent processing to a smaller region, with consequent savings in computational cost. Figure 9 shows the area of interest so computed from the image in figure 7.

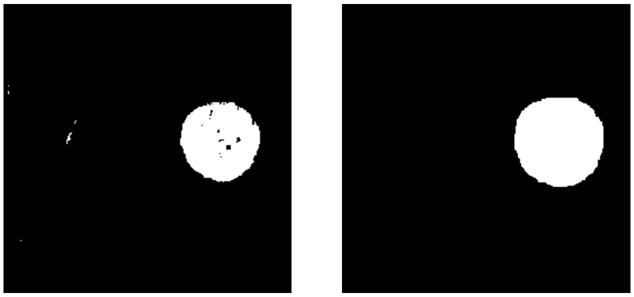

**Figure 8 a). Initial pupil mask, obtained by thresholding the image in figure 7 after highlight removal. b) Cleaned pupil mask, obtained from the mask in figure 8a by successive dilations and erosions.**

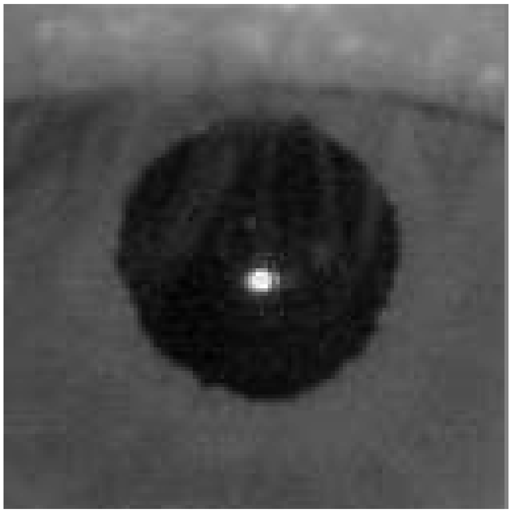

**Figure 9. 128x128 area of interest extracted from the image in figure 7 using the location of the pupil mask.**

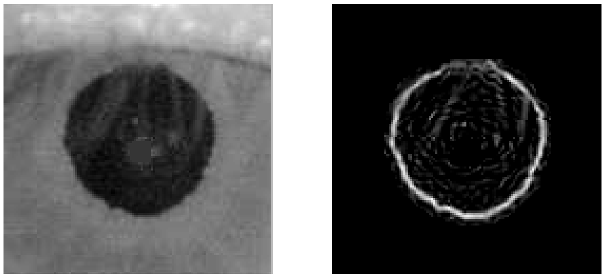

**Figure 10. a) Area of interest of highlight-removed image. b) Edge image, computed from the gradient of the image in (a), and masked by the pupil mask**

## 4.4 Pupil Edge Localization

A crude measure of pupil location is obtained from the pupil mask. We next extract the pupil edge by first computing gradients in the horizontal and vertical dimensions. At this point we might compute the gradient magnitude, but a similar but more sensitive measure is produced by computing the dot product of the gradient vector with a unit normal vector from the nominal center of the pupil to each point in the image (fig. 10b). This quantity is positive for dark-to-light transitions (such as the pupil margin we seek) but negative for light-to-dark transitions. By clipping from below at zero, we automatically reject all transitions of the wrong polarity.

We model the pupil margin signal as an ellipse having a Gaussian cross section. We solve for the optimal set of parameters using the iterative search procedure STEPIT [1]. We initially solve for the best fitting circle, by varying the position and radius. This is followed by a full 5 parameter fit in which two additional parameters are added for ellipse eccentricity and orientation. This process, by its iterative nature, is by far the most time-consuming of all the steps described in this paper.

## 4.5 Corneal Reflex Localization

The corneal reflex is quite conspicuous and relatively easy to find. The gradient magnitude of the unmasked image is computed (fig. 11a), and blurred (fig. 11b). The maximum of this image is located, and a mask is grown by dilation. Before computing the centroid, we must subtract off the average of the surrounding values, which are not zero. The mean surrounding value is determined by consideration of the pixels on the perimeter of the mask.

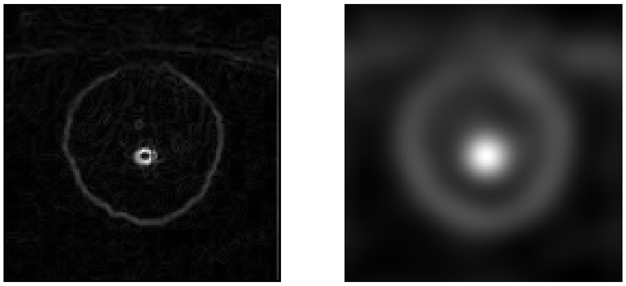

**Figure 11. a) Gradient magnitude of image in figure 9, used for corneal reflex localization. b) Blurred version used for mask construction.**

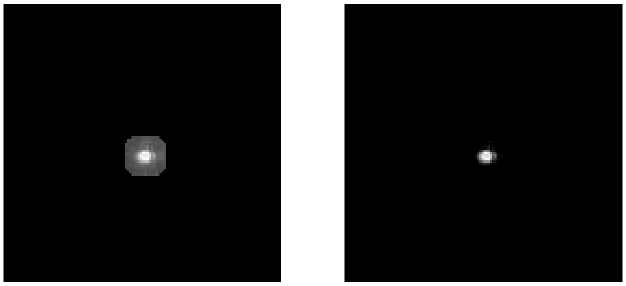

**Figure 12 a). Application of mask derived from the image in figure 11b, to the input image from figure 9. b) Image from figure 12a after background subtraction.**

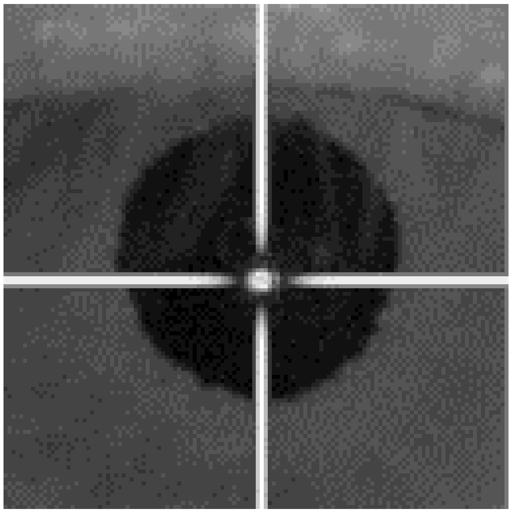

**Figure 13 Visualization of corneal reflex location obtained from the centroid of the image in figure 12b. Cross-hairs are Gaussian cross-section lines rendered with sub-pixel resolution.**

## 4.6 Blink Detection

A number of checks are performed at different points in the process to detect the occurrence of blinks. For example, as described above, the size and shape of the pupil mask are compared to predetermined limits; when the values fall outside of the legal range, subsequent processing on that frame is aborted, and a blink is signalled.

Measures such as these insure that ridiculous data are not generated from a bad frame, but are rather indirect evidence that a blink actually has occurred. A very simple measure that has been found to correlate quite well with the occurrence of a blink is the mean value of the original image. Although there are variations in mean value as gaze is shifted, the blinks leave a distinctive signature which is easily seen.

# 5. CALIBRATION

Calibration refers to the determination of how the measurements performed on images from the eye and scene cameras are related to the direction of gaze. We acquire a special calibration sequence before each experimental run. The calibration target consists of a 3x3 array of geometric shapes which are easy for the registration process to match unambiguously. A small fixation cross is presented in the center of each of the shapes in succession, and the subject is instructed to maintain fixation which making small head movements. We rely on the vestibulo-ocular reflex (VOR) which drives the eyes so as to cancel the effects of small head movements. We recover the head movements from the head tracking procedure described above, and by registering a model of the calibration image to the recorded images, we can determine the position in each scene camera image corresponding to the fixation target. Thus, we have a series of scene camera image locations and corresponding eye measurements. The singular value decomposition is used to solve for a least-squares solution of linear weights relating the measurements to scene camera coordinates. To recover the point of regard in the coordinate space of the display, an additional transformation relating scene camera coordinates to display coordinates (recovered from the head tracking procedure) is applied.

# 6. FUTURE WORK

The system described above works fairly well for situations in which the environment is a flat display screen whose content is known. Additional development is needed to deal with several situations of interest, such as dynamic screen content which is not necessarily known or reconstructable. An approach taken by some commercial systems is to embed reference markers in the monitor bezel; this is a reasonable approach, but requires that the scene camera have a field of view sufficiently large to capture the markers. This is at odds, however, with the goal of high precision, which favors a narrow field of view. A possible solution is a hybrid approach in which registration of bezel markers is performed along with frame-to-frame registration of captured scene content.

A more difficult problem is the use of scene camera images to locate the head for larger movements in three dimensional environments, such as flight simulator cabs, air traffic control towers, or a moving automobile. One approach to these situations might be the construction of a 3-D model of the environment which could then be used for registration.

# 7. ACKNOWLEDGMENTS

Thanks to Andy Hyatt and Mark Anderson for assistance with the video acquisition software, John Nguyen for development of the batch job dispatcher, Rick Jacoby for integration of polhemus code, and Valerie Heumer for help with preparation of the manuscript.

# 8. REFERENCES

[1] Chandler, J.P., STEPIT: finds local minima of a smooth function of several parameters. Behavioral Science (1986), 14, 81-82.

[2] Dougherty, E. R. An introduction of morphological image processing. SPIE (1992). Bellingham WA .

[3] Mulligan, J. B., Image processing for improved eye-tracking accuracy. Behavior Research Methods, Instruments and Computers (1996), 29, 54-65.

[4] Watson, A. B., Ideal shrinking and expansion of discrete sequences. NASA Technical Memorandum 88202 (1986), National Technical Information Service, Springfield VA,.