Limbus / Pupil Switching For Wearable Eye Tracking Under Variable Lighting Conditions

Wayne J. Ryan and Andrew T. Duchowski
School of Computing,
Clemson University, Clemson, SC 29634
{wryan | aduchow}@clemson.edu

Stan T. Birchfield
Department of Electrical and Computer Engineering,
Clemson University, Clemson, SC 29634
stb@clemson.edu

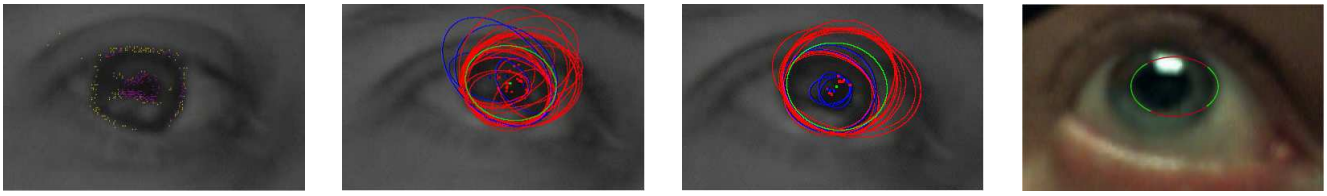


Figure 1: Ellipse fitting: detection of features on both pupil and iris boundaries; ellipses fit to sets of 5 randomly selected points; luminance-based delineation of features into two sets; proper detection of erroneous ellipse spanning both pupil and iris.

Abstract

We present a low-cost wearable eye tracker built from off-the-shelf components. Based on the open source openEyes project (the only other similar effort that we are aware of), our eye tracker operates in the visible spectrum and variable lighting conditions. The novelty of our approach rests in automatically switching between tracking the pupil/iris boundary in bright light to tracking the iris/sclera boundary (limbus) in dim light. Additional improvements include a semi-automatic procedure for calibrating the eye and scene cameras, as well as an automatic procedure for initializing the location of the pupil in the first image frame. The system is accurate to two degrees visual angle in both indoor and outdoor environments.

CR Categories: I.3.6 [Computer Graphics]: Methodology and Techniques—Ergonomics; J.4 [Computer Applications]: Social and Behavioral Sciences—Psychology.

Keywords: Wearable eye tracking, limbus tracking

1 Introduction

Wearable eye trackers allow collection of eye movements during the performance of natural tasks outside the laboratory, often allowing the use of unconstrained eye, head, and hand movements. Compared with the wealth of data obtained in laboratory settings, relatively little work has been done to collect eye movement data during the performance of such tasks, however. Current wearable trackers are scarce, expensive, and/or uncomfortable due to heavy headgear or equipment. Another serious limitation is their inability to handle variable lighting conditions, particularly in outdoor environments.

Figure 2: Our do-it-yourself wearable eye tracker from relatively inexpensive (∼$700) off-the-shelf components.

In this paper we describe a wearable eye tracker built with relatively inexpensive off-the-shelf components that is as comfortable to wear as a pair of plastic safety glasses (Figure 2). We are aware of only one other effort at development of a cheap, accurate, wearable eye tracker: the open source openEyes project [Li et al. 2006].

2 Previous Work

Modern eye tracking technology relies on cameras. Image processing methods are used to locate and track the pupil (and sometimes the corneal reflection of IR LEDs). In addition to tracking of the pupil, the corneal reflection (glint) of the auxiliary (IR) light source allows computation of the user's gaze point in the scene, if it can be reasonably assumed that the corneal glint is fixed with respect to the translating pupil. This is the technique employed by Babcock and Pelz's [2004] wearable eye tracker (an IR LED is positioned off-axis just to the left of the eye imaging camera). A laser diode is used to project points in the user's visual field to enable calibration of the tracker to the individual and subsequent Point Of Gaze (POG) calculation.

During calibration, the user is prompted to look at a series of points usually laid out in a grid pattern. Pupil and corneal glint features are recorded at each calibration point and used to fit a mapping from these features to the point of gaze in the scene.

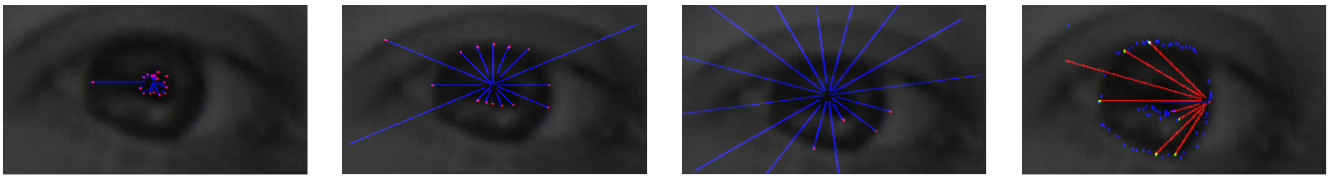


Figure 3: In the *Starburst* algorithm, rays are cast outward from an initial seed point assumed to lie within close proximity to the pupil, then cast back toward the center to generate additional points.

Infra-red illumination offers certain benefits due to the availability of an additional trackable feature (the corneal reflection). However, it may simultaneously pose a restriction on where the tracker can be used. For example, operation in sunlight has been problematic.

To allow operation in the visible spectrum, and to simplify complexity while reducing cost, Li and Parkhurst [2006] adopted the technique of limbus tracking (see below). Elimination of IR illumination simplifies the tracker's design by removing several components: the IR LED as well as a voltage regulator which required additional circuitry. Provided the headgear is sufficiently stable, any fixed point in the eye image frame can be used as an acceptable reference point for POG estimation. Although Li and Parkhurst's openEyes wearable eye tracker has matured to allow operation in the visible spectrum with an accuracy of about 1°, it is limited in several ways. Being a limbus tracker, it may not function particularly well under variable light conditions, i.e., in bright light when the contrast of the iris/sclera boundary is masked by specular reflections. Furthermore, user intervention is required at startup, e.g., the initial eye location is entered manually.

Our primary contribution is a mechanism for automatically switching between limbus and pupil/iris detection. Like Li and Parkhurst's eye tracker, ours also operates in the visible spectrum as well as in variable light conditions.

3 Pupil and Limbus Tracking

There are currently two basic approaches to POG calculation. The first is based on a 3D model of the eyeball and the computation of a ray emanating from a central point within the eye. The POG is then calculated (parametrically) as the intersection of the ray and some surface of interest in the environment (for example, see Hennessey et al. [2006]).

To develop a wearable eye tracker, Pelz et al. [2000] and then Li et al. [2006] took a second approach based on traditional video-oculography (VOG). The goal is to estimate (x,y), a vector from some static reference point in the eye image to the center of the limbus, with the limbus defined by an ellipse fit to the iris/sclera boundary (the vector will pivot as the eye rotates). To accomplish this, Li et al. [2005] proposed the *Starburst* algorithm, in which rays are projected from an initial starting point to detect pixels with high image gradients, then backprojected toward the center to increase the number of pixels prior to ellipse fitting. We adopt this as our starting point. Our approach differs, however, by automatically switching between tracking the edge of the pupil and tracking the limbus, allowing operation in variable light conditions. Processing a video frame involves three steps: image preprocessing, feature detection, and ellipse fitting.

Image Preprocessing. Like the *Starburst* algorithm, our approach begins by convolving the eye camera image with a Gaussian filter to reduce camera shot noise and lossy image compression artifacts resulting from the off-the-shelf camera. We combine Gaussian smoothing and differentiation into a single filter, enabling the computation of the image gradient with a single convolution. This is a simplification of the *Starburst* approach in which the gradient is only computed on an as-needed basis during feature detection.

Feature Detection. The purpose of feature detection is to locate pixels on the limbus or pupil boundary. These boundary points are found in two steps. As in *Starburst*, candidate feature points are found by casting rays out from an initial seed point and terminating the ray as it exits a dark region. We determine if the ray is exiting a dark region by checking the gradient magnitude and direction (see Figure 3). The feature detection process is then repeated with rays cast back from candidate boundary features toward the seed point, along rays within $\pm\phi$ degrees of the ray that generated the candidate boundary point (we set $\phi = 30$). This additional step tends to increase the ratio of the number of feature points on the desired boundary over the number of feature points not on the pupil contour [Li et al. 2006]. The ray does not terminate until the magnitude of the gradient component collinear with the ray exceeds some predetermined threshold. Although this technique is effective and efficient in some lighting conditions it is sensitive to the threshold chosen. Identification of an ideal threshold is confounded by the fact that higher thresholds are more effective in bright light, giving way to lower thresholds in dim light. Our algorithm iterates through multiple thresholds. Ellipses are fit to points generated at each threshold. Only the best ellipses are retained (see below).

Ellipse Fitting. The *Starburst* implementation uses Random Sample Consensus (RANSAC) [Fischler and Bolles 1981] to fit ellipses to the feature points, chosen for its tolerance to outliers, which are common in the observed feature sets. RANSAC proceeds in two steps: a population of ellipses is first generated, then the ellipses are evaluated to retain only the ones fitting best. Each ellipse is generated from five feature points selected at random. A conic section is fit to the selected points [Fitzgibbon et al. 1999]. Many such ellipses are created, and the number of feature points within some small epsilon of the ellipse is counted. The ellipse with the highest count is retained.

Our algorithm also utilizes a two step process. We generate random ellipses in a similar manner, but rather than evaluating them based upon a characteristic of the feature set, we evaluate them based upon characteristics of the original image. We label each pixel that the ellipse passes through as acceptable or not depending upon the magnitude and direction of the gradient at that pixel. The percentage of acceptable pixels is computed. The ellipse with the highest ratio is retained. This modification makes the algorithm more tolerant to poorly localized feature sets.

Starburst's ellipse fitting is further complicated by its inability to distinguish ellipses that partially span the pupil and those that partially span the limbus from those that exclusively adhere to one or the other. We suggest that the feature points be split into two groups based upon pixel luminance, simply partitioning them about the median value. Pixels on the pupil boundary are expected to exhibit lower luminance values than those on the limbus. This form of

(a) scene frame (b) low threshold (c) largest region (d) after flood fill (e) high threshold (f) final result

Figure 4: Automatic calibration depends on synchronization of eye and scene camera images and automatic detection of calibration dots based on thresholding.

feature point splitting allows the creation of two sets of ellipses, one corresponding to the iris/sclera limbus, the other to the pupil/iris boundary.

Figure 1 shows example results of feature detection. When ellipses are fit to all the points, erroneous ellipses are generated that span both the pupil and iris boundaries. By identifying and distinguishing between the points on the pupil and iris boundaries, such erroneous ellipses are less frequently created.

Switching between pupil and limbus is handled implicitly. The combination of multiple thresholds along with feature point splitting allows the algorithm to smoothly transition from tracking the pupil in bright light to tracking the limbus in dim light.

4 System

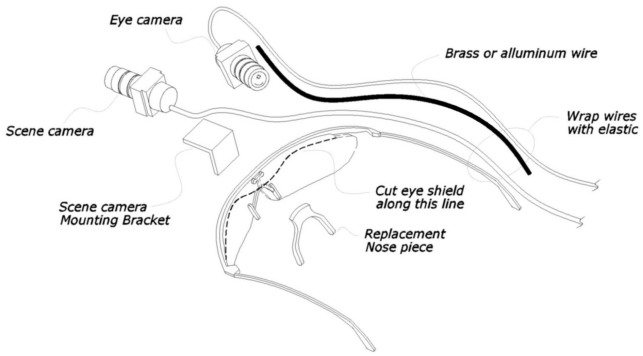


Figure 5: Eye tracker assembly.

Hardware. We use the same hardware design described by Li et al. [2006], with minimal modifications as necessary to facilitate material availability. The apparatus is constructed entirely from inexpensive commercial off-the-shelf (COTS) components. The simplicity of the design facilitates easy construction requiring only a minimal amount of expertise (see Figure 5). The entire parts list for the device include one pair of safety glasses (AOSafety X-Factor XF503), the more comfortable nose piece of a second pair of plastic sunglasses (AOSafety I-Riot 90714), black polyester braided elastic for wrapping the wires, two screws to connect the scene camera bracket and nose piece, a small aluminum or brass rod for mounting the eye camera, and two digital video minicams.

We use inexpensive digital video minicams (Camwear Model 200) from DejaView [Reich et al. 2004]. Each DejaView wearable digital mini-camcorder uses the NW901 MPEG-4 CODEC from Divio, Inc., enabling MPEG-4 video recording at 30 fps. Video is recorded on 512MB SD mini disks for offline processing.

Synchronization. The method requires the collection of two synchronized video sequences, one of the eye, the other of the scene being viewed. It cannot be assumed that the eye and scene cameras begin recording at exactly the same time. It is therefore necessary to synchronize the video streams of both cameras. As suggested by Li and Parkhurst [2006], a flash of light visible in both videos is used as a marker. We have employed an LCD monitor to display calibration dots, we therefore found it convenient to flash the monitor in order to create the short burst of light necessary for synchronization. We flash the monitor again to signal the end of calibration. The light from the monitor is sufficiently bright to be automatically identified in both video streams.

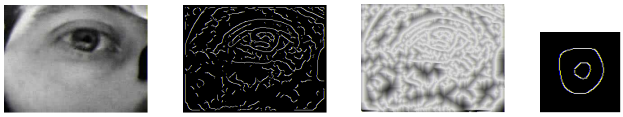
Calibration. We adopted the traditional video-oculography approach of calculating the point of gaze. In this approach, the image coordinates (x,y) of the center of the fitted ellipse are mapped to scene coordinates (s_x,s_y) using a second order polynomial [Morimoto and Mimica 2005]:

$$\begin{aligned} s_x &= a_0 + a_1x + a_2y + a_3xy + a_4x^2 + a_5y^2 \\ s_y &= b_0 + b_1x + b_2y + b_3xy + b_4x^2 + b_5y^2. \end{aligned}$$

Calibration requires the viewer to sequentially view a set of spatially distributed calibration points with known scene coordinates. This correspondence is used to compute the unknown parameters a_k and b_k via Lagrange's method of least squares.

The calibration points are displayed on a computer monitor and detected automatically in the scene camera image using a straightforward process illustrated in Figure 4. First, a low level threshold is applied to the image, and the largest dark region is assumed to be the computer screen (holes are eliminated in the largest region using a modified flood fill algorithm). A high-level threshold applied to the original image identifies candidate calibration points, and a logical AND operation applied to the computer screen image and the calibration points image yields the desired calibration point.

Initialization of Pupil Location. Even though the *Starburst* algorithm is able to accurately find the pupil center, it only performs a local search. It therefore needs a good starting point from which to begin searching. For frames in the middle of the video we may simply use the result from the previous frame, but for the first frame some other method must be devised. In contrast to the approach of Li et al. [2006], which requires manual entry of the initial start point, we have developed a fully automated solution.



(a) raw frame (b) Canny edges (c) chamfer image (d) pattern

Figure 6: Pre-processing steps for pupil/limbus pattern matching.

Our automatic localization of the pupil is based on the template matching algorithm of Borgefors [1988], which begins with the Canny [1983] edge detector on both the image to be searched and

a template of the object to be searched for. It then creates a chamfer image for which each pixel value is set to its distance from the nearest edge pixel. As suggested by Gavrila and Philomin [1999], a search map is then created by convolving the edge detected template with the chamfer image. The pixel in the search map with lowest value is the most likely location for the object.

One limitation of the standard chamfer algorithm is that it considers only the magnitude of the gradient of the image. Borrowing from the elliptical head tracking work of Birchfield [1998], we extend the algorithm to incorporate the direction of the gradient as well as the magnitude. Since the gradient is already computed by the Canny algorithm, the modification is straightforward. After inverting the chamfer image, we normalize the gradients for both the template and the search image, then augment the convolution operation with the dot product of the normalized gradient. Finally, the highest value in the search map is taken to be the most likely location.

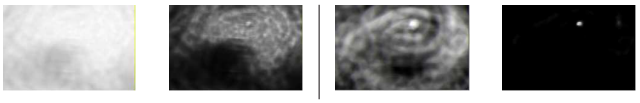


Figure 7: Effect of using the gradient direction to locate pixels associated with the pupil. The left image pair considers only edge distance, while the right pair also considers gradient direction.

5 Experimental Results

After applying a simple 5-tap FIR smoothing filter to the calculated gaze point, we display a simple cross hair at its location, as illustrated in Figure 8. Accuracy analysis refers to the gaze point after smoothing.

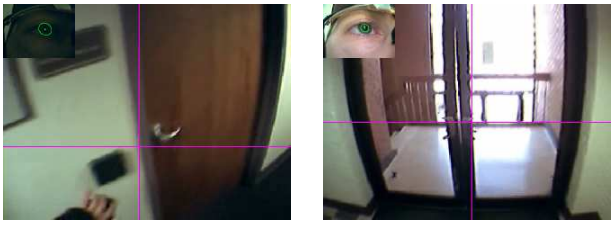


Figure 8: Cross hairs indicate point of gaze.

We evaluate our implementation by first calculating calibration coefficients. We then reprocess the same video, tracking the disparity between mapped gaze points and tracked calibration dots. We calculate the Euclidean distance between the mean calibration dot center and the mean gaze point over the period of gaze dot display. This distance is our error measurement given in pixels. The DejaView camera has approximately a 60° field of view, with video resolution of 320×240. Therefore a simple multiplication by 0.1875 converts our units of measurement from pixels to degrees visual angle. Using this metric, we were able to track the eye in three separate videos each with an average error less than 2°.

6 Conclusion

We have made three significant improvements upon previous work. The first two, initial pupil location detection and calibration dot tracking, improve ease of use by reducing the necessity for user intervention. Our third improvement, feature point luminance delineation, improves eye tracking versatility by allowing the algorithm to better distinguish between the pupil boundary and the limbus.

References

BABCOCK, J. S. AND PELZ, J. B. 2004. Building a Lightweight Eyetracking Headgear. In *Eye Tracking Research & Applications (ETRA) Symposium*. ACM, San Antonio, TX, 109–114.

BIRCHFIELD, S. 1998. Elliptical Head Tracking Using Intensity Gradients and Color Histograms. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 232–237.

BORGEFORS, G. 1988. Hierarchical Chamfer Matching: A Parametric Edge Matching Algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence 10*, 6, 849–865.

CANNY, J. F. 1983. Finding Edges and Lines in Images. Tech. Rep. AI-TR-720, MIT Artificial Intelligence Laboratory.

FISCHLER, M. A. AND BOLLES, R. C. 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM 24*, 6, 381–395.

FITZGIBBON, A., PILU, M., AND FISHER, R. 1999. Direct Least Square Fitting of Ellipses. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 21*, 5 (May), 476–480.

GAVRILA, D. M. AND PHILOMIN, V. 1999. Real-Time Object Detection for "Smart" Vehicles. In *International Conference on Computer Vision*. 87–93.

HENNESSEY, C., NOUREDDIN, B., AND LAWRENCE, P. 2006. A Single Camera Eye-Gaze Tracking System with Free Head Motion. In *ETRA '06: Proceedings of the 2006 Symposium on Eye Tracking Research & Applications (ETRA)*. ACM Press, New York, NY, 87–94.

LI, D., BABCOCK, J., AND PARKHURST, D. J. 2006. openEyes: A Low-Cost Head-Mounted Eye-Tracking Solution. In *Eye Tracking Research & Applications (ETRA) Symposium*. ACM, San Diego, CA.

LI, D. AND PARKHURST, D. 2006. Open-Source Software for Real-Time Visible-Spectrum Eye Tracking. In *Conference on Communication by Gaze Interaction (COGAIN)*. COGAIN, Turin, Italy, 18–20. URL: <http://www.cogain.org/cogain-2006/COGAIN2006_Proceedings.pdf>.

LI, D., WINFIELD, D., AND PARKHURST, D. J. 2005. Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches. In *Vision for Human-Computer Interaction Workshop (in conjunction with CVPR)*.

MORIMOTO, C. H. AND MIMICA, M. R. M. 2005. Eye Gaze Tracking Techniques for Interactive Applications. *Computer Vision and Image Understanding 98*, 4–24.

PELZ, J. B., CANOSA, R., AND BABCOCK, J. 2000. Extended Tasks Elicit Complex Eye Movement Patterns. In *Eye Tracking Research & Applications (ETRA) Symposium*. ACM, Palm Beach Gardens, FL, 37–43.

REICH, S., GOLDBERG, L., AND HUDEK, S. 2004. Deja View Camwear Model 100. In *CARPE'04: Proceedings of the 1st ACM Workshop on Continuous Archival and Retrieval of Personal Experiences*. ACM Press, New York, NY, 110–111.