Open-Source Software for Real-Time Visible-Spectrum Eye Tracking

Dongheng Li and Derrick Parkhurst
Human Computer Interaction Program
Iowa State University, USA
{dhli,derrick}@iastate.edu

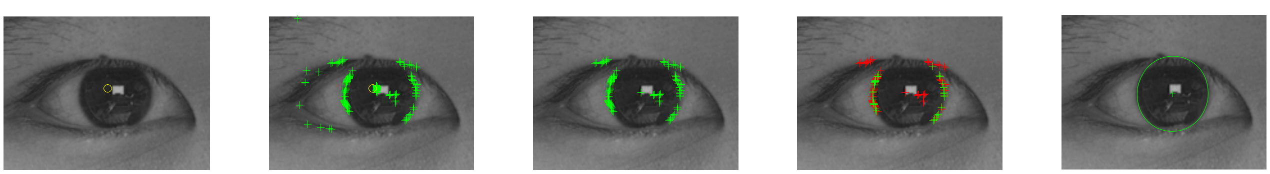
Keywords
Eye typing, limbus, RANSAC

Introduction

Video-based eye-tracking techniques frequently rely upon infrared-spectrum imaging because lighting and image exposure levels are precisely controllable through active illumination. However, there are a number of limitations to infrared-based eye tracking approaches. For example, infrared-spectrum systems perform poorly outdoors due to the presence of ambient infrared light. Moreover, performance of infrared-spectrum systems can vary significantly due to the individual differences in the physiological properties of the eye. For a more in depth review of these problems, see Hansen and Hansen (2005). We have developed a real-time eye-tracking system that uses visible-spectrum imaging in order to address these problems. We make this software freely available for download over the Internet as an open-source software package (see, http://hcvl.hci.iastate.edu/openEyes)

Algorithm

The most notable feature in visible-spectrum images of the eye is the limbus, i.e., the contour between the iris and the sclera (see Figure 1a). The position of the limbus is fixed with respect to the direction of gaze. The shape of the limbus in the image can be modeled as an ellipse. We adapted the Starburst algorithm (Li, Winfield, Parkhurst, 2005), originally designed to track the pupil in infrared-spectrum images of the eye, to track the limbus. In visible-spectrum eye tracking, light from ambient sources is relied upon illuminate the eye. Unfortunately, this can lead to the presence of uncontrolled specular reflections in the image of the eye. Fortunately, as the Starburst algorithm was originally designed to be robust to image noise, it is also well suited to handle the presence of extraneous specular reflections.



(a) (b) (c) (d) (e)

Figure 1. Eye-tracking algorithm (a) An eye image with the starting point shown in yellow circle. (b) detected features (green crosses). (c) Remaining features after distance filtering. (d) Inliers (green crosses) and outliers (red crosses) differentiated by RANSAC (e) Best fitting ellipse using only inliers.

Limbus Feature Detection

The algorithm begins at a starting point that is a best guess of the limbus center (see Figure 1a). This point is derived from the limbus center from the previous frame, and in the case of the first frame, is initialized as the center of the image. The limbus feature points are found by computing the derivatives along rays extending radially away from a starting point, until a threshold is exceeded. For each ray we detect two features before halting. An example set of detected features is shown in Figure 1b. Because the limbus is likely to be occluded by the eyelids and eyelashes, we restrict the directions of the rays. This range of angles is an adjustable to accommodate different users, but is initially taken to include -45^o to 45^o and 135^o to 225^o. One ray per degree of angle is traced resulting in at most 360 candidate limbus feature points.

Distance Filtering

Relying on the elliptical shape of the limbus in the images, a distance filter is applied to remove the features that are outliers. The features whose distance from the starting point is greater than 1.5 standard deviations from the mean are removed. The starting point is then replaced with the geometric center of the remaining features and the filtering is repeated again. The features remaining after filtering are shown in Figure 1c.

Ellipse Fitting

An ellipse is fitted to the candidate feature points using the Random Sample Consensus (RANSAC) paradigm (Fischler and Bolles, 1981). The candidate limbus feature points may still contain false alarms even after distance filtering, which would strongly influence the accuracy of the results if a least-squares fitting approach was used. RANSAC is an effective model estimation method in the presence of a large but unknown percentage of outliers in a measurement sample. We introduce two restrictions on the RANSAC fitting process to increase the robustness of the inlier selection process. First, only candidate ellipses with a radius ratio (major radius / minor radius) greater than 0.75 are considered. Second, only candidate ellipses of an area within plus or minus 1.5 standard deviations of the mean ellipse radius determined over the course of the tracking. The inliers are shown as green crosses and the outliers are shown as red crosses in Figure 1d. The final ellipse fit is shown in Figure 1e.

Application

We developed a low-cost remote eye tracker to test the algorithm. The remote eye tracker uses an inexpensive webcam, Unibrain Fire-i camera ($95 US dollars). The camera is mounted on the extended arm of a chin rest. To obtain a full-frame image of the eye we replace the original lens with a 12mm zoom lens ($70 US dollars) which required the addition of a CS lens mount ($10 US dollars). The system requires the user to places his/her head in the chin rest to assure proper alignment of the camera and use the remote eye tracker for desktop applications. The distance between the eye and the display was 26 inches and the visual angle of the screen was 32 degrees.

To calculate the point of gaze in the scene image, a mapping between the limbus center and the point of gaze must be determined. The user is therefore required to look at a 9-point grid, for which the scene locations are known. We use then can estimate a second-order polynomial mapping. After calibration, the user's point of gaze in the scene for any frame can then be established from the limbus center using this mapping. The average error in terms of visual angle is approximately 1 degree of visual angle after calibration. A significant limitation of this remote eye tracker is that it requires the user to hold very still to avoid introducing error into the estimated point of gaze. With a head-mounted system, the user would not be similarly restricted.

Conclusion

We developed a visible-spectrum eye-tracking algorithm and tested this algorithm with a remote eye-tracking system. We have made this algorithm freely downloadable and open source so that it can be easily integrated into human computer interaction applications, for example, gaze-based communication. We believe that in order for applications of eye tracking to become widespread, low-cost eye tracking solutions must be developed. With the decreasing cost of hardware and now the availability of free, open-source eye-tracking software, we expect that gaze-based interfaces will become more prevalent.

References

Hansen, D. and Hansen, J. (2005), Review of Current Camera-based Eye Trackers, *First Conference on Communication by Gaze Interaction*, 2005, pp 7-9.

Fischler, M. and Bolles, R., Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, *Communications of the ACM*, vol. 24, 1981, pp. 381-395.

Li, D., Winfield, D., Parkhurst, D. J. (2005). Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches. Proceedings of the IEEE Vision for Human-Computer Interaction Workshop at CVPR, 1-8.