Proceedings of the 29th Annual International
Conference of the IEEE EMBS
Cité Internationale, Lyon, France
August 23-26, 2007.

A High Speed Eye Tracking System with Robust Pupil Center Estimation Algorithm

Xindian Long[1], Ozan K. Tonguz[1], and Alex Kiderman[2]

Abstract— **This paper presents a new high-speed head-mounted binocular eye position measurement system using infrared lighting and image processing technology. Current eye tracking systems either run on-line at a lower speed, do the processing off-line, or use dedicated hardware to reach high on-line processing rates. The eye position measurement system we developed only uses a general-purpose computing system and off-the-shelf imaging devices. The binocular system can provide on-line horizontal and vertical measurement at a speed of 150 Hz on a desktop system with a 3 GHz Pentium processor. We report a novel dual-mode software system and a two-step processing algorithm to increase the processing rate. A symmetric mass center algorithm is designed to provide more accurate measurements when the pupil area is partially occluded. Using synthetic eye images, we show that our algorithm performs consistently better than some of the widely used algorithms in industry.**

I. INTRODUCTION

Robust and non-invasive eye tracking is an important component in many applications including human computer interaction, virtual reality, driver assistance, and diagnosis or early screening of some health problems. Abnormal eye movement can be an indication of diseases in balance disorder, Diabetic Retinopathy, Strabismus, Cerebral Palsy, Multiple Sclerosis, etc. A video-based eye tracking system is non-invasive compared to other methods including scleral search coil systems [1]. Head mounted eye tracking systems are more accurate than remote video-based systems, electro-oculography (EOG) systems [2], cornea reflection systems [3], or flying-spot laser based systems [4]. The temporal resolution of the video-based eye tracking systems used to be limited by the camera speed and computation power. With the availability of ever increasing computation power, the development of digital camera and image processing technology, it is now possible to exploit the full potential of the video-based eye tracking systems.

To capture high speed eye movement in real time, Clarke presents a system at a sampling rate of 400 Hz by using smart sensors and a custom-designed DSP/FPGA architecture for pre-processing [5]. The SMI system reaches a speed of 500 Hz and also involves dedicated hardware [6]. The specialized hardware design induces considerable development cost and time.

This paper presents a high speed head-mounted binocular eye tracking system we have recently developed at Carnegie Mellon University. The herein reported results clearly indicate that it is possible to measure the eye movement at a speed of 150 Hz by just using a general purpose computer system. The system utilizes off-axis infrared lighting and the black pupil technique so that the pupil area can be easily extracted. To detect the pupil center at high speed, the system employs a dual-working-mode mechanism and a two-step processing algorithm. First, an approximate pupil location is searched in a low resolution image, then the image area around the approximate location is processed and an accurate pupil center is calculated. We observed that droopy eye lids partially covering the pupil is a quite common phenomenon. Hence, a *symmetric mass center* algorithm based on geometric properties of ellipses is developed to achieve a more accurate measurement when the eye lid is partially occluded.

II. RELATED WORK

Detecting the pupil is the most frequently used method to track the horizontal and vertical eye position [7]-[9]. Unfortunately, most early pupil detection systems make the oversimplifying assumption that the pupil is circular, and that the pupil center is midway between the horizontal and vertical extents. In practice, even a circular pupil appears elliptical at eccentric eye position. Zhu et al. proposed to use the curvature characteristics of the pupil boundary and fit them to an ellipse [8]. This technique gives a more robust and accurate estimation of the pupil center. Starburst is an eye tracking system that uses a hybrid algorithm combining feature based (edge point extraction) and model-based approaches (ellipse fitting) [9]. However, the computation power these methods require adversely affect the processing rate.

The approach presented in this paper for finding pupil center is an improvement over the naive *center of mass* algorithm. The symmetric mass center algorithm we developed gives a more robust measurement for the eye position. Processing frame rate is enhanced by eliminating the bottleneck in transferring and processing of the entire image. Locating the approximate position of the pupil at a low-resolution image can significantly decrease the region that needs processing for the feature extraction step. Our algorithm improves the accuracy compared with the mass center algorithm while maintaining the high speed operation.

III. SYSTEM OVERVIEW

Table I shows the main components of the system. Figure 1 shows the goggle (the high speed eye tracking system)

[1] Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213-3890, USA.
{xlong, tonguz}@ece.cmu.edu
[2] Neuro Kinetics, Inc., 128 Gamma Drive, Pittsburgh, PA, 15238, USA.
akiderman@neuro-kinetics.com

1-4244-0788-5/07/$20.00 ©2007 IEEE

TABLE I: System Configuration

Component Name	Specification	Number
Computer Desktop System	Pentium IV 3 GHz CPU 1G RAM, 800MHz bus	1
Software Platform	Windows XP Pro, Labview 7.1	1
Interface Card	IEEE 1394b Fireware, 2 Port	1
High Speed Camera	Point Grey DragonFly Express	2

with the two cameras installed. Two hot mirrors are used to reflect the eye images to the cameras and at the same time allow a broad view angle for patients. Infrared LED is used for illumination and infrared pass filters are installed before the cameras to exclude light from the environment. Two cables connect the cameras to the IEEE 1394b Fireware card installed in the computer desktop system.

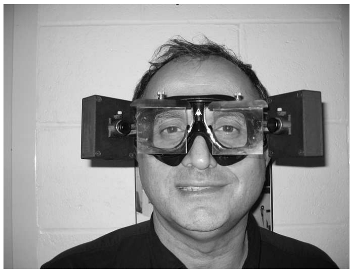


Fig. 1: A subject wearing the goggle with two cameras installed

The images captured by the cameras are transfered to the computer. The image processing software we developed extracts the pupil and calculates its geometric center. From the pupil center, the horizontal (θ) and vertical(ϕ) rotation angle of the eyes are calculated according to Eq. 1.

$$\begin{array}{rl} \text{vertical rotation angle } \phi: & sin(\phi) = \frac{-dy}{z} \\ & cos(\phi) = \sqrt{1 - sin^2\phi} \\ \text{horizontal rotation angle } \theta: & sin(\theta) = \frac{dx}{zcos(\phi)} \\ z = \sqrt{R_e^2 - r^2} & \end{array} \tag{1}$$

where, r is the pupil radius, R_e represents the radius of the eye ball and it is obtained during calibration procedure according to the method described in [10], and dx, dy are the deviation in the pupil center location relative to the reference position when the eye is looking straight ahead.

The software first works in the low speed mode when full size images are captured and displayed so that the operator can select the region of interest (ROI); then the software can be switched to the high speed mode when only the partial image inside the ROI window is captured. In the high speed mode, a two-step processing algorithm first processes the image in the ROI window at a low pixel density resolution to find the approximate location of the pupil. After that, a tiny trace window containing the entire pupil area is defined and further processing is done in the trace window to calculate the precise location of the pupil center. The system increases the processing rate for horizontal and vertical eye position measurement up to 150 Hz on a Pentium IV 3 GHz computer for both cameras, displaying measurement results and captured eye images on the interface. It achieves this high speed by eliminating the bottleneck caused by capturing and processing the full size images without sacrificing convenience and spatial resolution.

IV. High Speed Pupil Detection

A. Existing Pupil Detection Method

In a typical pupil detection procedure based on the *center of mass* algorithm, the captured image is first transformed into a binary image by a user-set threshold. Everything under the threshold is labeled as “one” (considered object pixels) and everything else is labeled as “zero” (considered as the background). Then, one can use a blob analysis algorithm such as the Labview Virtual Instrument (VI), IMAQ Complex Particle, to find objects (consisting of connecting object pixels) in the image. Finally, the software chooses the largest object, which is the pupil, and calculates the geometric center of the object.

A 320×240 pixel Region of Interest (ROI) window is usually sufficient to capture the whole dynamics of eye movement assuming the eye is right at the center of the ROI window when the subject is looking straight ahead. However, it is difficult to adjust the wear-on goggle to make sure that the eye is right at the center of the captured image area. Therefore, it is necessary to capture a larger image area and then set the ROI window for processing. In addition to the processing time, computation cost of capturing the image, user interface monitoring, and image displaying also slow down the overall frame rate.

The aforementioned factors limit the performance of the binocular system. Currently, if the full size images are captured and processed directly, the final system processing rate (image capturing and interface updating time included) one can achieve is about 80 Hz on a Pentium IV computer. We aim to find a software solution to increase the processing rate.

B. The Dual-Mode Capturing Software and the Two-step Processing Algorithm

To enhance the processing rate, we observed that the size of the image captured and processed is the key to the problem. The pupil itself is pretty small, typically less than 120×120 pixels, but it moves around. If one can find the approximate location of the pupil first, one can then process a much smaller image to find the more accurate location of the pupil center by processing a much smaller image.

We developed a new dual-mode capturing software with a *two-step* processing algorithm so that it can achieve a high frame rate without sacrificing the accuracy. We first capture and process the images at full size and full resolution (640×480) as described in section IV-A. The operators can set the threshold, adjust the ROI window in this low speed mode. After that, the user can switch to the high speed mode. In the high speed mode, the system captures the partial image in the ROI window specified above. This makes the Fireware 1394b interface board capable of transmitting images at a higher frame rate and reduces the processing time as well.

To reach an even higher speed, we developed an algorithm to first track the approximate position of the pupil and then determine a tiny image window enclosing the pupil for further processing. The image is down sampled at $\frac{1}{4} * \frac{1}{4}$ rate of the original resolution and the procedure described in section IV-A is used to locate the approximate pupil center. Then, the system specifies a tiny trace window slightly larger than the detected pupil area so that we do not lose any pixels. We process the tiny image at full pixel density resolution using the symmetric mass center algorithm described below to locate the accurate position of the pupil center.

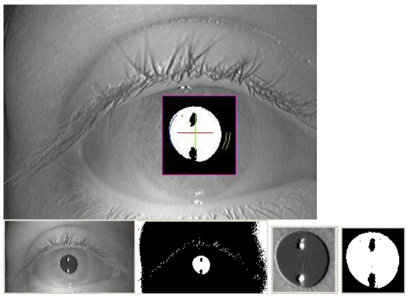


Fig. 2: The upper row shows a full pixel density resolution grey level image (the partial image defined by the ROI window), with the small threshold image inlayed on the right location. The lower row shows, from left to right, the down sampled grey level image, the threshold image of the down sampled image, the grey level image of full pixel density resolution in the tiny trace window, and the small threshold image in the tiny trace window.

Figure 2 shows the grey level images and threshold images in different resolution. “Object” pixels close to the image borders are caused by camera distortion; they can be eliminated by the simple criteria of being connected to the borders, and this step can be completed in the loww speed mode. By implementing the dual-mode capturing and the two-step processing algorithm described above, the current software reaches a final system speed of 150 Hz (binocular system) in the high speed mode, including proper image and data display on the user interface.

V. The symmetric Mass Center Algorithm

In the second step, we extract the image in the tracing rectangle at full pixel density resolution and apply a new symmetric mass center algorithm to calculate the pupil center coordinates. The center of mass algorithm assumes that the pupil area can be approximated by a circle or an ellipse. This assumption will be violated when the eye lid covers part of the pupil area and it leads to large measurement error. The symmetric mass center algorithm utilizes only the non-occluded, symmetric portion of the pupil area to estimate the pupil center.

Suppose that an ellipse is partially occluded as shown in Figure 3, the mass center of the non-occluded area will not yield the correct ellipse center. From geometric properties of ellipses, we know that if we find the maximum parallelogram $S_1E_1S_3E_3$, the mass centre of the area enclosed by $\overset{\frown}{S_1S_3}\overset{\frown}{E_3E_1}$ is a good estimate of the ellipse centre, given a significant area of the ellipse is not occluded.

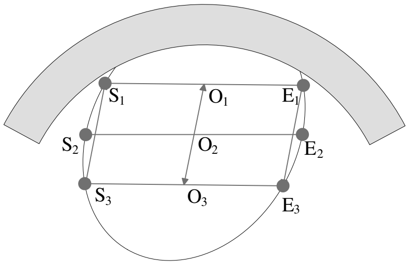


Fig. 3: Locating the centre for a ellipse that is partially occluded

TABLE II: Pseudo code for the symmetric mass center algorithm

```
     Main Program
1)   $Im_t$ = downSample ($Im_{in}$)
2)   ($px, py, rect, Im_s$) = NMassCentre ($Im_t$)
3)   (ox, oy) = SymmetricMassCentre ($Im_s, px, py$)

     SymmetricMassCentre ($Im_s, px, py$)
1)   $line_u = py$, $line_d = py$
2)   { $S_u, E_u$ } = scanImgLine ($Im_s, line_u$)
     $len_u = |S_u E_u|$
3)   { $S_d, E_d$ } = scanImgLine ($Im_s, line_d$)
     $len_d = |S_d E_d|$
4)   If ($len_u > len_d$)
        $line_u = line_u - 1$,
     else
        $line_d = line_d + 1$
5)   If $len_u$ and $len_d$ are decreasing, and $S_u E_u S_d E_d$ is no
     longer close to a parallelogram
        break to step 7
6)   Loop back to step 2.
7)   $(ox, oy)$ = PMassCenter ($Im_s, line_u, line_d$)
```

We designed our symmetric mass center algorithm according to this principle, and the pseudo code is shown in Table II. In the code, *NMassCentre* is the procedure that implements the naive mass center algorithm, *scanImgLine* is a simple procedure for finding the starting and ending object pixel in the image row, and *PMassCenter* finds the mass center of the partial object falling between row $line_u$ and row $line_d$ ($line_u$ and $line_d$ represent respectively the top and bottom row of the “symmetric” part of the ellipse, and are obtained iteratively in the SymmetricMassCentre procedure).

VI. Results

A. Speed of horizontal and vertical position measurement

We tested the processing rate of the software with a Pentium IV 3 GHz computer. The data is obtained without including the processing time for capturing the image from the camera and the time for user interface updating.

Figure 4 shows the result. The solid line with dots shows the processing rate (for single image) in the low speed mode. The solid line with asterisks shows the processing rate (for single image) in the high speed mode. The x axis is the normalized image size in the ROI window defined. Table III lists the normalized image sizes and the corresponding dimensions of the ROI window. Although this is not the final system speed, it shows the effect of employing the two-step processing algorithm. Observe that the new algorithm introduced a considerable performance improvement in terms of processing rate.

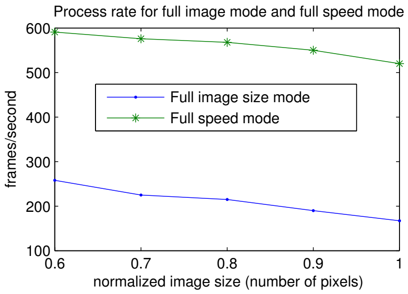


Fig. 4: Processing rate in different ROI size

TABLE III: Dimensions of the ROI window and the corresponding normalized image size

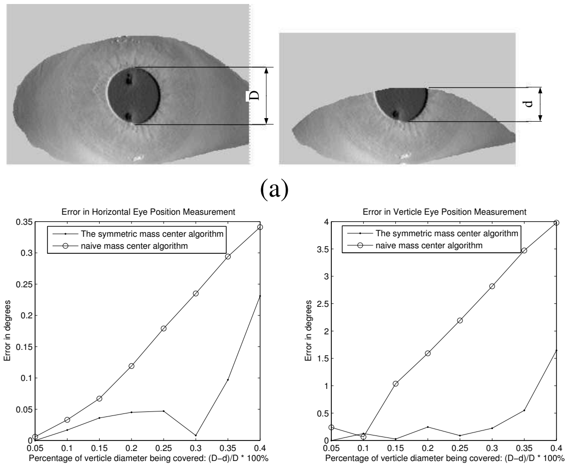
ROI size	400 × 300	380 × 250	340 × 260	335 × 250	310 × 230
Normalized size	1	0.9	0.8	0.7	0.6

B. Accuracy of the symmetric mass centre algorithm

In this test, we compared the accuracy of measured eye position for different algorithms by putting the eye at the centre position, and artificially varying the degrees that the pupil area is occluded. Figure 5-(a) shows an eye with the full pupil area viewable on the left and the eye that is partially occluded on the right. D is the length of the vertical diameter and d is the length of the un-occluded portion of the diameter. In the test, we will vary the *degree of occlusion* defined as $\frac{D-d}{D}$. Experiments are performed to compare two different algorithms: the symmetric mass centre algorithm and the original centre of mass algorithm. In Figure 5-(b), the left figure shows the error in the horizontal measurements, and the right figure shows the error in vertical measurements. The plot shows that when the degree of occlusion increases, the error for both algorithm increases. In most cases, the symmetric mass centre algorithm performs better than the naive mass centre algorithm. We can also observe that when the degree of occlusion is less then 0.3, the error that resulted from occlusion for the symmetric mass centre algorithm is moderate (less than $0.5°$), while the error for the naive mass centre algorithm increases linearly with the degree of occlusion.

VII. Conclusion and future work

While many eye-tracking systems have been developed before, we have developed a new video-based eye tracking system with high temporal resolution using only general-purpose computing and image capturing devices. More specifically, in this paper, we have presented a high-speed eye position measurement system that does not sacrifice either accuracy or convenience. The high-speed 2D eye position measurement system achieves a binocular processing rate of 150 Hz for horizontal and vertical eye measurement, including necessary results and video displaying on the interface. The system employs a dual-mode capturing mechanism so that only the necessary part of the entire image is captured but at the same time maintains the convenience of operation. To calculate the pupil center, we designed a two-step processing algorithm; i.e., first processing a down-sampled image to find the approximate location of the pupil and then processing the full resolution image in the small window at the approximate pupil location. This method improved the processing rate to about $230\% - 300\%$ of the original method.



(a)

(b)

Fig. 5: Measurement results for different algorithms when pupil area is occluded: (a) shows the vertical diameter D and the un-occluded portion of it, d; (b) plots the measurement results for different algorithms in terms of the different portion of the vertical diameter occluded.

References

[1] J. M. Furman, H. Collewijn, T. C. Jansen, and A. V. Van Den Berg, "Human gaze stability in the horizontal, vertical and torsionaldirection during voluntary head movements, evaluated with a three-dimensional scleral induction coil technique", in *Vision Research*, vol. 27, no. 5, pp. 811-828, 1987.

[2] A. Duchowski, "Eye-Based Interaction in Graphical Systems: Theory and Practice" available at *http://vret.ces.clemson.edu/sigcourse/notes/c05-TOC.pdf*, Clemson University.

[3] S. Shih, Y. Wu, and J. Liu, "A calibration-free gaze tracking technique", in *Proceedings of the 15th international conference on pattern recognition*, Barcelona, Spain, vol. 4, pp. 4201-4205, 2000.

[4] K. Irie, B. A. Wilson, R. D. Jones, P. J. Bones, and T. J. Anderson, "laser-based eye-tracking system", in *Behavior Research Methods, Instruments, & Computers*, vol. 34, no. 4, pp. 561-572, Nov., 2002.

[5] A. H. Clarke, C. Steineke, and H. Emanuel, "High image rate eye movement measurement : A new generation of tracker using cmos image sensors and dedicated fpga processing", in *Proceedings 3d VOG Workshop Tubingen*, pp .12-16, Nov. 30-Dec. 2, 1999.

[6] Sensor Motoric Instruments, "Fastest Video-Based Eye Tracking", available at http://www.smi.de/.

[7] Z. Zhu, K. Fujimura, and Q. Ji, "Real-time eye detection and tracking under variuos light conditions". in *Proceedings of the 2002 symposium on Eye tracking research and applications, ACM Press*, pp. 139-144, 2002.

[8] D. Zhu, S. T. Moore, and T. Raphan, "Robust pupil center detection using a curvature algorithm", in *Computer methods and programs in biomedicine*, vol. 59, no. 3, pp. 145-57, 1999.

[9] L. Dongheng, D. Winfield, and D. J. Parkhurst, "Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches", in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2005)*, vol. 3, pp. 79, San Diego, USA, June 2005.

[10] S. T. Moore, T. Haslwanter, I. S. Curthoys, and S. T. Smith, "A geometric basis for measurement of three-dimensional eye position using image processing", in *Vision research*, vol. 36, pp. 445-459, 1996.