ELSEVIER

Computer Methods and Programs in Biomedicine 59 (1999) 145–157

Computer Methods and Programs in Biomedicine

Robust pupil center detection using a curvature algorithm

Danjie Zhu [a], Steven T. Moore [a,b], Theodore Raphan [a,b,*]

[a] *Department of Computer and Information Science, Brooklyn College of the City University of New York, 2900 Bedford Avenue, Brooklyn, NY 11210, USA*

[b] *Department of Neurology, Mt. Sinai School of Medicine, New York, NY 10029, USA*

Received 31 March 1998; received in revised form 23 November 1998; accepted 25 November 1998

Abstract

Determining the pupil center is fundamental for calculating eye orientation in video-based systems. Existing techniques are error prone and not robust because eyelids, eyelashes, corneal reflections or shadows in many instances occlude the pupil. We have developed a new algorithm which utilizes curvature characteristics of the pupil boundary to eliminate these artifacts. Pupil center is computed based solely on points related to the pupil boundary. For each boundary point, a curvature value is computed. Occlusion of the boundary induces characteristic peaks in the curvature function. Curvature values for normal pupil sizes were determined and a threshold was found which together with heuristics discriminated normal from abnormal curvature. Remaining boundary points were fit with an ellipse using a least squares error criterion. The center of the ellipse is an estimate of the pupil center. This technique is robust and accurately estimates pupil center with less than 40% of the pupil boundary points visible. © 1999 Elsevier Science Ireland Ltd. All rights reserved.

Keywords: Eye movements; Video; VOG; Image processing; VOR

1. Introduction

Accurate eye movement recording and monitoring in three dimensions (3D) has been growing in importance in the differential diagnosis of diseases associated with vertigo and other neurological disorders [1]. Determining the accuracy and trajectory of rapid eye movements provides information about how well individuals can localize targets of interest [2–4], as well as the function of neural circuits in the paramedian pontine reticular formation (PPRF) [5] and superior colliculus (SC) [6]. Accurate determination of eye orientation is also important for assessing the gain and phase of the angular vestibulo-occular reflex (aVOR), which precisely compensates for head movements to maintain the eyes stable in space for visual

* Corresponding author. Tel.: +1-718-9514193; fax: +1-718-9514489.

E-mail address: raphan@sci.brooklyn.cuny.edu (T. Raphan)

PII: S0169-2607(98)00105-9

acuity while moving [7–9]. A non-invasive measure of eye orientation is to track the pupil center relative to the head. If this can be done accurately and robustly, then eye orientation in 3D can be determined from the pupil center coordinates and the torsional component of eye position (i.e. rotation of the eye about the line of sight, or visual axis). Thus, an accurate determination of pupil center is important for determining eye orientation in 3D, which in turn impacts on how well we can study the normal and patho-physiology of disorders of the central nervous system. Accurate tracking of the pupil center is also important in a number of other areas, including virtual reality, reading, perception, industrial design, and aids for the disabled.

A number of systems have recently been developed which use video imaging of the pupil (video-oculography or VOG) to determine 3D eye orientation [10–19]. These systems operate at video sampling rates of up to 60 Hz, requiring that the processing be automated due to the large volume of data generated during testing. In the majority of VOG methodologies, it is assumed that the optic axis (the line through the pupil center and the center of eye rotation) coincides with the visual axis or direction of gaze [20]. Horizontal and vertical eye position can therefore be determined from a computation of the pupil center. Accurate determination of the pupil center is crucial for finding the torsional component of eye orientation, since it is generally computed using the grey-level intensity information of the iris (iral signatures) obtained from a circular iral sampling annulus centered on the pupil. Cross-correlation of iral signatures obtained from test images with a reference signature provides an estimate of torsional eye position [10,12–16]. Improved accuracy of torsional computation has been achieved using algorithms which compensate for the eccentricity of the iral sampling annulus according to eye orientation [16], but this technique is still highly dependent on an accurate determination of the pupil center. Even a small error in the pupil center coordinates corresponding to 0.2 mm can induce errors in the torsional measurement of up to 2° [21]. Given the small magnitude of torsional eye movements (typically less than 6°) such errors are significant.

The center of mass algorithm is a commonly used procedure in VOG systems, which for fully visible, focused (ideal) images of the pupil provides an accurate estimate of the pupil center [12,13,15,16,19]. Significant errors in the pupil center computation arise due to artifacts from eyelids, eyelashes, shadows and corneal reflections, which occlude the pupil boundary. Some attempts have previously been made to increase the robustness of pupil center detection. Low-pass filtering and other image processing techniques have been used to remove the effect of ‘holes’ in the pupil image due to corneal reflections [19]. The pupillary chord technique [14,22] can also tolerate corneal reflections inside the pupil boundary. The accuracy of both techniques, however, is degraded if the corneal reflection lies on the pupil boundary. A clipped circular disk model for the pupil has been developed that allows an estimation of the pupil center when obscured by the upper eyelid, but does not account for obscuration of the lower pupil edge [23]. In addition, the pupil boundary is elliptical when the eye is in eccentric gaze positions. While finding the pupil center using a circular fit can be accurate in many instances, it is inaccurate in estimating the center when the eye is at eccentric gaze positions and the pupil is partially occluded (see section 2.5 for a more complete discussion).

We have developed an algorithm that uses the curvature characteristics of the pupil boundary to determine the visible portion of the pupil, which provides improved estimates of the pupil center when it is obscured by a host of artifacts. The curvature algorithm discriminates between edge points that lie on the smooth pupil boundary and those that lie on the intersection of the pupil with eyelids, eyelashes, corneal reflections or shadows. The non-occluded boundary points are used as input to an ellipse-fitting procedure that provides a robust estimate of the pupil center. We have studied the performance of this algorithm using images of both artificial and human eyes.

2. Pupil center algorithm

The algorithm for finding the pupil center operates on a 2D digitized image of the eye. The algorithm is divided into a number of logical steps, which are described in detail below.

2.1. Pupil boundary detection algorithm

The pupil boundary was determined by first identifying all points related to the pupil, i.e. defining a pupil 'blob.' The pupil blob was generated from the 256 grey level image (Fig. 1(A)) using a threshold operation. Pixels with grey levels below the threshold value were assigned to a black blob (grey level of 0) and all others were designated as white 'background' pixels (grey level of 255), resulting in a binary image (Fig. 1(B)). This image usually contained multiple blobs due to artifacts such as non-uniform lighting conditions or eyelashes and shadows. Each blob was isolated and labeled using 8-neighbour connectivity analysis, which classifies pixels according to their relationship with the 8 surrounding pixels [24]. The blob associated with the pupil was then identified as having the largest area (based on the number of pixels) and a small standard deviation of the interior points from the blob's center of mass. All other spurious blobs were eliminated (Fig. 1(C)) and the resultant binary image of the pupil was smoothed using a 2D median filter, which removed isolated and noisy edge points without blurring the pupil boundary. The algorithm was optimized such that filtering was only performed on regions with abrupt transitions in grey level, i.e. near the pupil edge points.

The contour following algorithm was initialized by first calculating the center of mass, P_c, of the

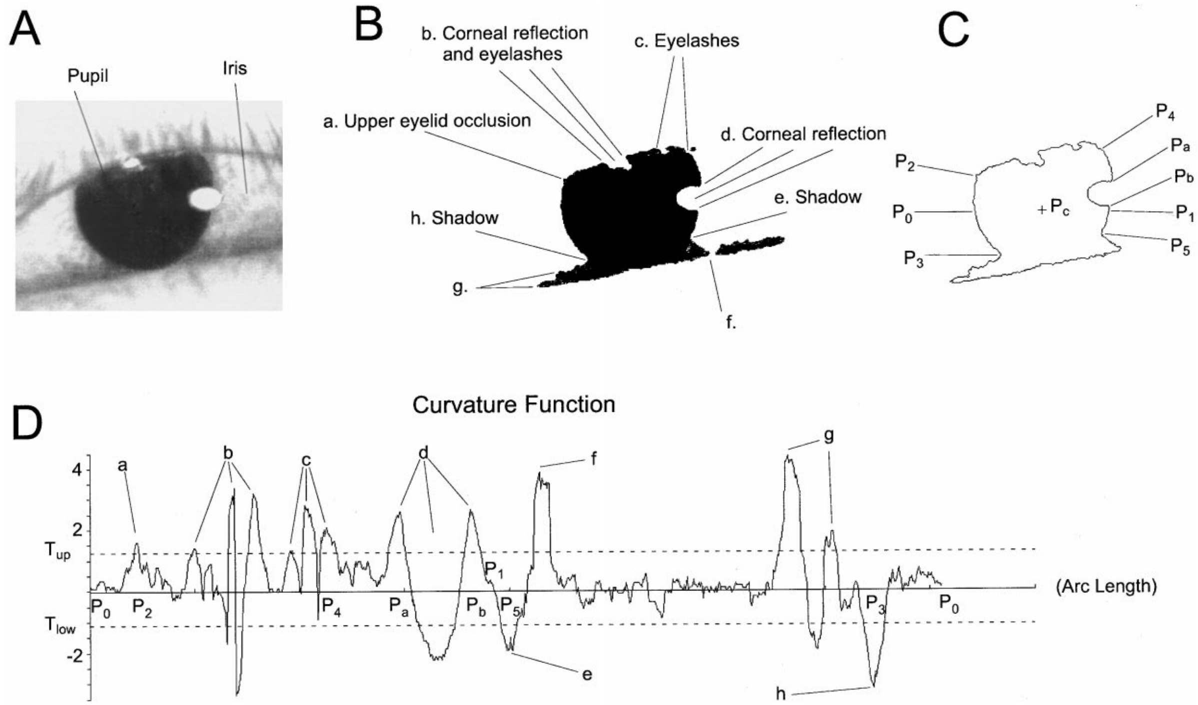

Fig. 1. Derivation of the curvature function: (A) Original 256 grey level image of a human eye in total darkness. Note that due to the infra-red illumination of the eye the pupil appears black, whereas the iris is much lighter; (B) Binary image of the pupil 'blob'; (C) Boundary points of the pupil 'blob'; (D) Curvature function of the pupil boundary points. The upper and lower curvature thresholds (T_{up} and T_{low}) define the range of normal pupil curvature. The large positive and negative peaks outside of this range indicate occlusion of the pupil boundary by various artifacts.

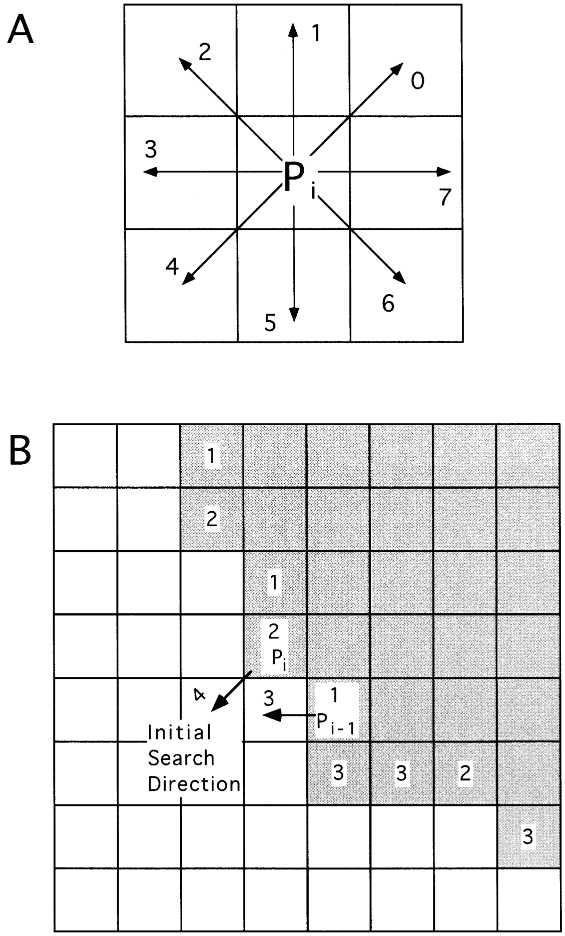

Fig. 2. Direction coding of the pupil boundary: (A) Each edge point is assigned a direction code relative to the previous point; (B) Segment of the pupil boundary showing direction codes. The initial search directions for points P_i and P_{i-1}, are shown. See text for details.

pupil blob (Fig. 1(C)). Two edge points, P_0 and P_1, were then found by searching horizontally to the left and right of P_c until a white background pixel was encountered. While searching for the two edge points, any corneal reflections inside the pupil were identified and skipped using a width criterion (~ 30 points), i.e. any holes within the pupil blob were classified as a corneal reflection if the hole width was less than this value. Starting from P_0, each boundary point P_i was assigned an integer code from 0 to 7, representing its direction relative to the previous point P_{i-1}, [24] (Fig. 2(A)). The current point P_i was located by scanning the 8-connected neighbors of P_{i-1}, in a clockwise direction until the first black pixel was encountered (Fig. 2(B)). The starting point for this search was the neighboring point whose direction code was orthogonal to the direction code of P_{i-1}, and exterior to the pupil blob (Fig. 2(B)). The contour following algorithm continued in a clockwise direction until the pupil boundary was closed (Fig. 1(C)). The edge points may include points on the intersection of the pupil with the eyelids, eyelashes, corneal reflections and shadows.

2.2. Curvature calculation

To facilitate an accurate ellipse fit to the pupil boundary, it was necessary to separate the visible pupil boundary from edge points lying on eyelids, eyelashes, corneal reflections and shadows. This was accomplished by using a heuristically based curvature algorithm. The curvature κ of an arc is defined as the rate of change of the angle (defined as the angle of the tangent to the boundary curve at a particular point with respect to the X-axis) with respect to the arc length s (Fig. 3(A)) [25].

$$\kappa = \frac{d\alpha}{ds} \approx \frac{\Delta\alpha}{\Delta s} \qquad (1)$$

A discrete form of Eq. (1) was used to calculate the curvature of the pupil boundary. For each point P_i, the curvature was obtained from two windows placed on either side of P_i (Fig. 3(B)), with P_i included in the first window. A 'direction histogram' was calculated for each window (usually containing ten boundary points), and the average of the four most frequently occurring directions was used as an approximation for the angle. This estimate minimized the effect of noisy edge pixels. The difference between the average values of the two adjacent windows was then divided by the window length (a discrete form of Δs), which was defined as the curvature value of the edge point P_i. This process was repeated for each boundary point giving the curvature function (Fig. 1(D)).

2.3. Peak transitional point detection

The curvature function for the non-occluded pupil boundary is a small positive value, which is dependent on the size of the pupil in the image. Typical values for our imaging set-up ranged from 0.3 for fully dilated pupils to 1.3 for constricted pupils with minor fluctuations due to digitization noise. Based on these values, upper and lower thresholds were defined for curvature values for pupils within the normal physiological range, with thresholds ranging from ± 1.0 to 2.0. A sudden deviation of the curvature from these threshold values (i.e. a peak in the positive or negative direction) indicates an occlusion of the pupil boundary by an eyelid, eyelashes, corneal refection or shadow. The peak transitional point (local maximum or minimum) could be used to determine the location of the obstruction (Fig. 1(D)). Correct setting of the curvature threshold was crucial for obtaining accurate pupil center estimates. If the curvature threshold is set too high, the peaks associated with pupil occlusion are not detected, and boundary points associated with artifacts are included in the ellipse fit. For example, if the threshold is set to ± 5.0, all boundary points on the pupil blob would be included. If the threshold is set too low, points on the smooth pupil boundary would be eliminated. In both cases, the accuracy of ellipse fitting would be reduced.

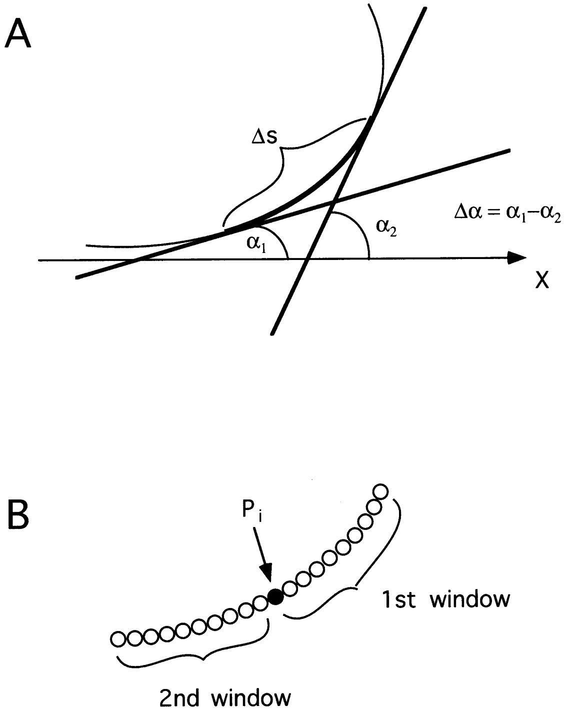

Fig. 3. Calculation of the curvature for a pupil edge point: (A) Definition of curvature for a continuos curve; (B) A discrete estimate of the curvature of the pupil boundary at a particular point P_i is determined using two windows. See text for details.

A curvature-scanning program has been developed which calculated all the peak transitional points in the curvature function for a particular pupil blob. We could now define heuristics to segment the pupil boundary into regions occluded by artifacts, and non-occluded portions that will be used in finding the pupil center.

2.4. Pupil boundary segmentation algorithm

Various heuristics were developed which could be applied to the curvature function to segment the non-occluded pupil boundary from artifacts. These heuristics were based on observations of the curvature characteristics of pupil boundaries when occluded by eyelids, eyelashes, corneal reflections and shadows (Fig. 1(A)):

1. Eyelids are characterized by isolated breakpoints with positive peaks that exceed a distance threshold ($D_L = 20$ points) from their nearest neighbor breakpoints, both in terms of arc length (Fig. 1(D), a) and Euclidean distance (Fig. 1(B), a). The positive peaks arise due to the convexity of the pupil and overlying eyelid (Fig. 1(A)), and the fact that the curvature function is computed in a clockwise direction.
2. Eyelashes are characterized by consecutive breakpoints which are closely spaced within a certain threshold (D_L) in both arc length (Fig. 1(D), b, c) and Euclidean distance (Fig. 1(B), b, c).
3. Corneal reflections are characterized by negative peaks bracketed by positive peaks on either side due to their concave nature (Fig. 1(D), d). The distance between the two positive peaks in terms of arc length (Fig. 1(D), d)

is much larger than the Euclidean distance between the two breakpoints (Fig. 1B, d) which falls within the range D_L–D_H (where $D_H = 40$ points).

4. Shadows are characterized by isolated breakpoints with negative peaks that exceed a distance threshold (D_H) from their nearest neighbor breakpoints, both in terms of arc length (Fig. 1(D), e, h) and Euclidean distance (Fig. 1(B), e, h). The negative peaks arise because of the concavity of the shadow–pupil intersection (Fig. 1(B), e, h).

The distance thresholds D_L and D_H were determined by processing a large database of images of a range of human eyes. The actual value is highly dependent on the scale factor of the eye image, which for our set-up was approximately 55 pixels per mm. For different magnifications, the distance thresholds would need to be scaled appropriately. If D_L is set too low, eyelashes will be recognized as eyelids or corneal reflections. If it is too high, eyelids and corneal reflections will be classified as eyelashes. This would not affect the accuracy of the segmentation algorithm, as all points associated with eyelids, eyelashes and corneal reflections are excluded from the ellipse fit. If D_H is not set correctly, corneal reflections and shadows can be confused. This poses a potential problem, as all boundary points associated with corneal reflection should be eliminated from the ellipse fit, whereas points on one side of the shadow/pupil intersection should be retained (Fig. 1(B), h–a).

Edge points on the smooth pupil boundary had a small curvature value. For example, the smooth pupil boundaries from P_2 to P_3, P_4 to P_a and P_b to P$_5$ (Fig. 1(C)) correspond to the curvature function values between segments labeled ‘h’–‘a’, ‘c’ (rightmost peak) to ‘d’ (leftmost peak) and ‘d’ (rightmost peak) to ‘e’ (Fig. 1D). Shadows can also produce this effect (Fig. 1(D) ‘f’–‘g’), but are eliminated from the extraction process using the heuristic rules described below.

For a typical pupil image, the boundaries to the left and right of the center of mass of the pupil, P_c, are visible, except when corneal reflections occlude these regions (Fig. 1(C)), or if the eye is closed. Boundary segments above and below P_c are typically associated with occlusion by eyelids, eyelashes or shadows. The aim of the pupil boundary segmentation algorithm was to search for break points that separate the visible pupil boundaries located to the left and right of P_c from any artifacts located above and below P_c. Starting from the two pupil edge points P_0 and P_1, the algorithm searched for the four breakpoints P_2, P_3, P_4 and P_5 (Fig. 1(C)). If a segment of corneal reflection boundary was encountered (P_aP_b in Fig. 1(C)), it was skipped and the search continued for the next breakpoint. P_2 and P_4 do not exist if the upper eyelid or eyelashes do not occlude the upper part of the pupil boundary. If the lower eyelid or shadows do not occlude the lower part of the pupil boundary, P_3 and P_5 do not exist.

Using the above heuristics, the breakpoints P_2, P_3, P_4 and P_5 were found and the non-occluded boundary segments P_2P_3, P_4P_a and P_bP_5 (Fig. 1(C)) determined using the following criteria:

1. P_2 is the first breakpoint after P_0 in a clockwise direction that satisfies condition i, ii or iii.
2. P_3 is the first breakpoint before P_0 in a clockwise direction that satisfies condition i, ii or iii.
3. P_4 is the first breakpoint before P_1, in a clockwise direction that satisfies condition i, ii or iii.
4. P_5 is the first breakpoint after P_1, in a clockwise direction that satisfies condition i, ii or iii.

where conditions i, ii and iii are as follows:

i. shadow/pupil intersection point, or
ii. eyelashes/pupil intersection point, or
iii. breakpoint not associated with corneal reflections.

2.5. Ellipse fit to pupil boundary points

Fitting a circle to the pupil boundary [23] provides an accurate estimate of the pupil center for horizontal and vertical eye positions up to ± 30° when all pupil boundary points are visible. A circle-fit is less accurate, however, when the pupil is partially occluded and in an eccentric position. We have verified this using a simulated pupil orientated 30° horizontally and 30° vertically. A circle-fit generated errors in the horizontal and vertical position greater than 0.5° when more than 15% of the pupil boundary was occluded. The error rose to 1° when 25% of the boundary was covered. In contrast, an ellipse fit generated negligible error (less than 0.01°) when 15% of the pupil boundary was obscured. The overall errors were less than 0.5° for up to 68% of pupil boundary occlusion. On the basis of these results, an ellipse-fit was used to obtain estimates of the pupil center.

Ellipse-fitting algorithms compute the best-fit ellipse using some linear least squares error criterion [26]. These algorithms are simple and accurate when sufficient data points are given, which closely describe the ellipse. Standard linear least squares techniques may fail when the data is sparse or very noisy. To overcome this problem, geometric algorithms have been introduced which minimize the sum of the squares of the geometric distances to the given points [27]. These geometric algorithms are much more computationally expensive (by a factor of 10–100) compared to the linear least squares algorithms [27].

We compared the performance of the linear least squares approach and the geometric approach of Gander et al. [27], and found that the pupil center estimated from both techniques were identical. This was due to the fact that noisy data points (artifacts) on the pupil boundary had been removed using the heuristics described above, and the number of data points was large (150–500 points) and closely described the pupil boundary. Furthermore, the geometric algorithm was more than 100 times slower than the linear least squares technique, and was not suitable for real-time application. On this basis we used the standard least squares error approach to estimate the parameters for the best-fit ellipse using Singular Value Decomposition [26]. From these parameters, the pupil center could then be estimated (see Appendix), and the horizontal and vertical components of the eye position calculated. For a detailed description of the derivation of the Euler angles representing a 2D eye position from the pupil center image coordinates, see [16].

3. Experimental methods

Video images of the eye were obtained using a commercial miniature CCD camera mounted on modified diving goggles (EYECAM, Iscan Inc., Cambridge MA). The eye was illuminated by an infrared (IR) diode and the image reflected onto the CCD by an IR-sensitive dichroic mirror. The resultant video image was digitized into a 640 × 480 array with 256 grey level resolution by a frame grabber (PCI-IMAQ-1408, National Instruments Corp., Austin TX) and processed on an IBM PC-compatible computer with a Pentium-Pro 200 MHz processor.

Images of an artificial eye were also obtained to assess the accuracy of the curvature algorithm in determining the pupil center of partially occluded pupils. The artificial eye consisted of a white plastic sphere having dimensions similar to that of a human eyeball (radius 16 mm). A 5 mm diameter black circle was fixed to this sphere to act as a pupil. A pattern of radial lines was placed around the pupil to simulate a human iris. The artificial eye was positioned such that the center of the sphere coincided with the center of a Fick gimbal. Rotation of the artificial eye was possible about the horizontal and vertical axes using verniers with an accuracy of 0.1°. The gimbal and video camera were solidly mounted on an aluminum bench and were aligned such that the 'optic axis' of the artificial eye was aligned with the normal of the image plane of the camera, which corresponded to the reference position. The torsional position of the camera was adjusted so that horizontal or vertical rotations of the gimbal corresponded to horizontal or vertical movements of the artificial eye in the image plane.

4. Results

4.1. Artificial eye

The accuracy of the curvature algorithm in determining the pupil center was plotted as a function of boundary occlusion from an image of an artificial eye. Two cases commonly encountered with human eyes were simulated: symmetric occlusion from the upper and lower eyelids (Fig. 4(A)), and upper eyelid occlusion only (Fig. 4(B)). The artificial eye was placed in an eccentric position with Euler angles of + 10° yaw + 10° pitch, which generated an elliptical projection of the pupil onto the image plane, and falls within the range of typical human eye movements. The pupil

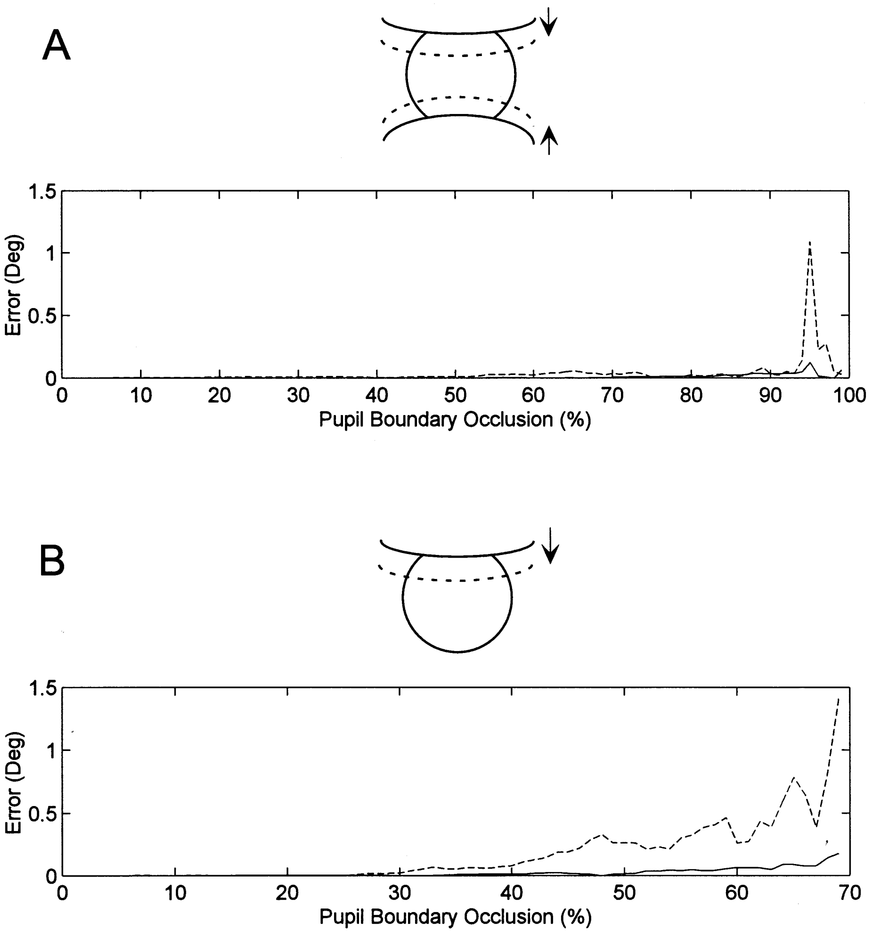

Fig. 4. Results of pupil center detection for an artificial eye as a function of occlusion. The absolute value of the error in eye position (degrees) in yaw (solid line) and pitch (dashed line) is plotted as a function of the percentage of occlusion of the pupil boundary: (A) Symmetric occlusion of the pupil from upper and lower eyelids; (B) Asymmetric occlusion from the upper eyelid only. Note that due to the asymmetric nature of the occlusion, the errors are significantly larger than for symmetric occlusion.

boundary points were obtained from an image of this artificial eye using the techniques described above. Portions of the pupil boundary array were then deleted in 1% increments from both the upper and lower regions (Fig. 4(A) insert), or the upper region only (Fig. 4(B) insert), and the center of the pupil was computed using the remaining points (see Appendix A). The position of the pupil center in Euler angles was then calculated [16], and the absolute value of the error (in degrees) was determined as a function of occlusion.

For symmetric occlusion of the upper and lower regions of the pupil (Fig. 4(A)), the absolute value of the error in yaw and pitch pupil position was less than 0.5° for up to 95% occlusion (which we consider the limit of acceptable accuracy for pupil center calculation). The errors induced in pupil position for upper eyelid occlusion were considerably greater, due to the asymmetrical nature of the occlusion (Fig. 4(B)). The error in yaw and pitch position was less than 0.5° for up to 65% occlusion.

4.2. Real eyes

We have tested the curvature algorithm on images of human eyes with a wide range of eyelid, eyelash, corneal reflection and shadow occlusion. Four examples are shown below (Figs. 5 and 6). A pupil with upper eyelid occlusion of 41% is shown in Fig. 5(A). The resultant binary image from the threshold operation (Fig. 5(B)) shows the clipped upper region of the pupil due to the eyelid. The result of the curvature algorithm (Fig. 5(C)) demonstrated that the visible portions of the pupil boundary were identified (solid line), while the segment occluded by the eyelid (dotted line) was eliminated from the ellipse-fitting routine. The resultant ellipse was superimposed on the original image (Fig. 5(D)), and overlay the pupil boundary. The accuracy of this ellipse fit was consistent with the error bounds established using the artificial eye above, i.e. occlusion of the pupil boundary of up to 40% results in an error of less than 0.1° (Fig. 4). Ellipse fit results for a large pupil obscured by a shadow (Fig. 6(A)), corneal reflection (Fig. 6(B)), and upper eyelash occlusion of 42% (Fig. 6(C)) demonstrated the accuracy of the curvature algorithm in the presence of a range of artifacts.

5. Discussion

The main contribution of this study was the development of an algorithm to accurately calculate the pupil center in the presence of common artifacts such as eyelids, eyelashes, corneal reflections and shadows. This algorithm will also improve the accuracy of 3D eye position measurement, as robust pupil center detection is a critical step in the determination of occular torsion about the visual axis. The novel aspect of this

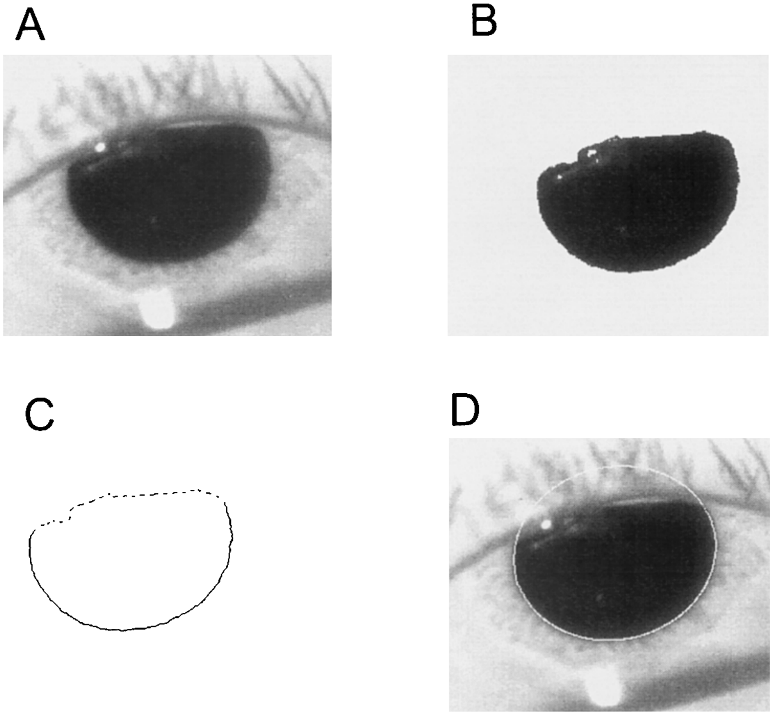

Fig. 5. Results of the curvature algorithm for a pupil occluded by the upper eyelid. The subject was in total darkness, which resulted in a large pupil diameter: (A) Original 256 grey level image; (B) Binary image of the pupil blob; (C) Results of the boundary segmentation algorithm. Visible portions of the pupil (solid line) have been extracted from the occluded portions (dashed line); (D) Best-fit ellipse to the visible pupil boundary points overlays the actual pupil.

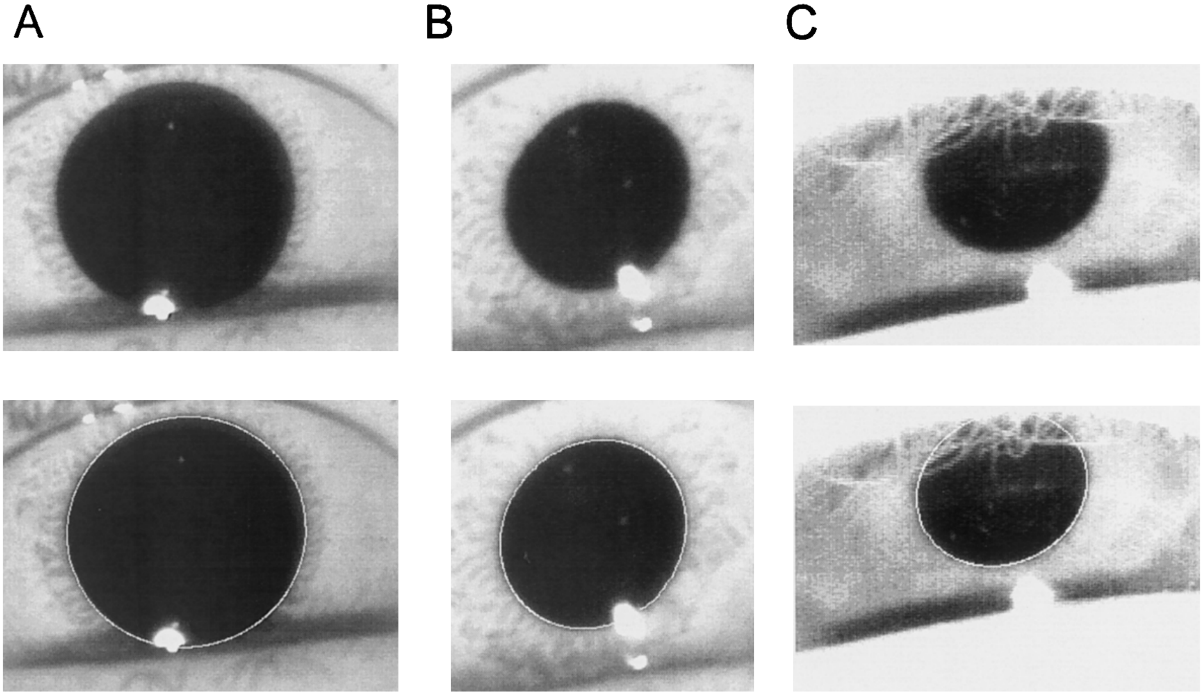

Fig. 6. Results of the curvature algorithm showing original images and the best-fit ellipses to the visible pupil boundary points overlaying the actual pupils. Pupils are occluded by: (A) shadow; (B) corneal reflection; and (C) upper eyelid and eyelashes.

work was the utilization of the curvature characteristics of the pupil boundary. This is in contrast to previous pupil center finding algorithms that have utilized edge coordinate averaging (barycenter) [10,28–30], pupillary chords [14,22] and the calculation of the center of mass of the pupil blob [12,13,15,16,19]. This curvature approach may also be useful in other image processing applications which involve the labeling of boundary segments of any geometrical shape in the presence of noise.

The essence of the curvature algorithm developed in this study is the utilization of the curvature characteristics of the pupil boundary to segment the edge points into visible and occluded portions of the pupil. The curvature function of the pupil boundary eliminates artifacts using the heuristic that curvature values for a smooth pupil boundary segment correspond to small constant values within a narrow identifiable range. Occlusion of the pupil boundary by eyelids, eyelashes, corneal reflections or shadows induces atypical curvature patterns in the curvature function of positive or negative peaks outside the range of smooth pupil boundaries. The remarkable aspect of the algorithm is that from the characteristics of these peaks, simple heuristics can be derived to detect the intersections of the smooth pupil boundary with the artifacts. The visible pupil edge points are then readily extracted.

Another essential aspect of the algorithm is the use of an ellipse fit to the non-occluded edge points using a least squares error criterion. The fitting algorithm uses an ellipse as the model base of the pupil boundary. The robustness of the curvature algorithm lies in the fact that, through application of the heuristics to the curvature function to segment the pupil edge points, there is a high probability that all points used in the ellipse fitting routine actually lie on the visible pupil boundary. This ensures that an accurate and robust pupil center estimate is provided by the center of the ellipse, even when only 40% of the pupil boundary is visible.

A potential problem with the curvature algorithm occurs when the obscuration of the pupil by an eyelid is small and the curvature transition from the visible pupil to the eyelid is smooth, i.e. is within the threshold range for a smooth pupil boundary. Even if such an artifact is erroneously included as input to the ellipse fitting routine, the fact that this algorithm constrains the pupil boundary to an ellipse has a tendency to cancel such errors. Moreover, the curvature algorithm will give no worse result than that of existing pupil detection algorithms in the presence of occlusions. Another shortcoming of the present algorithm is that it uses fixed values for the upper and lower curvature thresholds for the smooth pupil boundary. As the curvature value for the pupil boundary varies with pupil diameter, a more robust segmentation of the pupil boundary could be obtained by incorporation of an adaptive curvature threshold based on the pupil size from previous images.

VOG is the most viable non-invasive technique for measuring eye movements in three dimensions, and is a potentially important tool in human clinical and research studies. Existing VOG techniques, however, are inaccurate when the pupil is partially obscured by a variety of commonly occurring artifacts. In order for VOG to become the measurement technology of choice, it is necessary that pupil detection algorithms be capable of dealing with such artifacts. While much work needs to be done to attain the goal of a practical, robust video-based 3D eye movement tracker, the curvature algorithm for estimating the pupil center provides an important step towards achieving this goal.

Acknowledgements

Supported by NIH grants EY 04148, DC 03284, NASA grant through NSBRI, New York State HEAT grant, and PSC-CUNY grant 664230. We would also like to thank Drs Bernard Cohen and George Wolberg for helpful comments.

Appendix A. Determination of the center of an ellipse

In general, the projection of the pupil boundary onto the image plane is an ellipse. The center of the best-fit ellipse to the pupil boundary corresponds to the projection of the center of the pupil onto the image plane, and can be used to derive the relative position of the optical axis of the eye in 3D head coordinates [16]. The coordinates of the center of the ellipse can be calculated directly from the parameters of the equation of the ellipse as follows.

Any ellipse in the image plane can be represented as a function of the image coordinates (x, y)

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0 \tag{A1}$$

The parameters $\{A, B, C, D, E, \text{and } F\}$ can be derived from a best-fit ellipse to the visible pupil boundary points. Letting α be the angle between the major axis of the ellipse and X-axis of the image plane, we have [25]:

$$\alpha = \frac{\arctan\left(\dfrac{B}{A - C}\right)}{2} \tag{A2}$$

In order to simplify the calculation of the pupil center, we perform a rotation transformation on the $X-Y$ coordinate frame, such that the major and minor axes of the ellipse are parallel to the rotated axes $X'-Y'$, respectively. The coordinates of the ellipse in $X-Y$ space (x, y) are related the rotated coordinates (x', y') by

$$x = x' \cos\alpha - y' \sin\alpha \tag{A3}$$

$$y = x' \sin\alpha + y' \cos\alpha$$

By substituting Eq. (A3) into Eq. (A1), the ellipse in the rotated coordinate frame $X'-Y'$ can be represented as

$$\frac{\left(x' + \dfrac{D'}{2A'}\right)^2}{\dfrac{F''}{A'}} + \frac{\left(y' + \dfrac{E'}{2C''}\right)^2}{\dfrac{F''}{C'}} = 1 \tag{A4}$$

where:

$$A' = A\cos^2\alpha + B\cos\alpha\sin\alpha + C\sin^2\alpha \tag{A5}$$

$$C' = A\sin^2\alpha - B\cos\alpha\sin\alpha + C\cos^2\alpha$$

$$D' = D\cos\alpha + E\sin\alpha$$

$$E' = -D\sin\alpha + E\cos\alpha$$

$$F' = F$$

$$F'' = \frac{D'^2}{4A'} + \frac{E'^2}{4C'} - F'$$

From Eq. (A4), the center of the ellipse in the rotated image plane (X′–Y′) is: (−D′/(2A′), −E′/(2C′)). Using (A3) we can find the coordinates of the center of the ellipse in the original X-Y image plane:

$$x_0 = -\frac{D'}{2A'}\cos\alpha + \frac{E'}{2C'}\sin\alpha \quad \text{(A6)}$$

$$y_0 = -\frac{D'}{2A'}\sin\alpha - \frac{E'}{2C'}\cos\alpha$$

References

[1] J.R. Leigh, D.S. Zee, The Neurology of Eye Movements, Davis, Philadelphia, 1983.

[2] K. Hepp, V. Henn, Spatio-temporal recording of rapid eye movement signals in the monkey paramedian pontine reticular formation (PPRF), Exp. Brain Res. 52 (1983) 105–120.

[3] D. Tweed, T. Vilis, Implications of rotational kinematics for the oculomotor system in three dimensions, J. Neurophysiol. 58 (1987) 832–849.

[4] C. Schnabolk, T. Raphan, Modeling three dimensional velocity-to-position transformation in oculomotor control, J. Neurophysiol. 71 (1994) 623–638.

[5] V. Henn, B. Cohen, Coding of information about rapid eye movements in the pontine reticular formation of alert monkeys, Brain Res. 108 (1976) 307–325.

[6] D.A. Robinson, Eye movements evoked by collicular stimulation in the alert monkey, Vis. Res. 12 (1972) 1795–1808.

[7] T. Raphan, V. Matsuo, B. Cohen, Velocity storage in the vestibulo-ocular reflex arc (VOR), Exp. Brain Res. 35 (1979) 229–248.

[8] T. Raphan, B. Cohen, How does the vestibulo-ocular reflex work, in: R.W. Baloh, G.M. Halmagyi (Eds.), Disorders of the Vestibular System, Oxford University Press, New York, 1996, pp. 20–47.

[9] S.B. Yakushin, M.J. Dai, J.-I. Suzuki, T. Raphan, B. Cohen, Semicircular canal contribution to the three-dimensional vestibulo-ocular reflex: A model-based approach, J. Neurophysiol. 74 (1995) 2722–2738.

[10] M. Hatamian, D.J. Anderson, Design considerations for a real-time occular counterroll instrument, IEEE Trans. Biomed. Eng. BME 30 (1983) 278–288.

[11] J.A. Parker, R.V. Kenyon, L.R. Young, Measurement of torsion from multitemporal images of the eye using digital signal processing techniques, IEEE Trans. Biomed. Eng. BME 32 (1985) 28–36.

[12] T. Vieville, D. Masse, Occular counter-rolling during active head tilting in humans, Acta Otolaryngol. 103 (1987) 280–290.

[13] A.H. Clarke, W. Teiwes, H. Scherer, Video-oculography—an alternative method for measurement of three-dimensional eye movements, in: R. Schmid, D. Zambarbieri (Eds.), Oculomotor Control and Cognitive Processes, Elsevier Science Publishers B. V., North-Holland, 1991, pp. 431–443.

[14] A.H. Clarke, Image processing techniques for the measurement of eye movement, in: J. Ygge, G. Lennerstrand (Eds.), Eye Movements in Reading, Pergamon, Oxford, 1994, pp. 21–38.

[15] S.T. Moore, I.S. Curthoys, S.G. McCoy, VTM–an image-processing system for measuring ocular torsion, Comput. Methods Program Biomed. 35 (1991) 219–230.

[16] S.T. Moore, T. Haslwanter, I.S. Curthoys, S.T. Smith, A geometric basis for measurement of three-dimensional eye position using image processing, Vis. Res. 36 (1996) 445–459.

[17] J.E. Bos, B. de Graaf, Ocular torsion quantification with video images, IEEE Trans. Biomed. Eng. BME 41 (1995) 351–357.

[18] E. Groen, J.E. Bos, P.F.M. Nacken, B. de Graaf, Determination of ocular torsion by means of automatic pattern recognition, IEEE Trans. Biomed. Eng. BME 43 (1996) 471–478.

[19] J.B. Mulligan, Image processing for improved eye tracking accuracy, Behav. Res. Methods, Instrum. Comput. 29 (1997) 54–65.

[20] L.R. Young, D. Sheena, Methods & designs: Survey of eye movement recording methods, Behav. Res. Methods Instrum. 7 (1975) 397–429.

[21] T. Haslwanter, S.T. Moore, A theoretical analysis of three-dimensional eye position measurement using polar cross-correlation, IEEE Trans. Biomed. Eng. BME 42 (1995) 1053–1061.

[22] J. Barbur, W.D. Thomson, P.M. Forsyth, A new system for the simultaneous measurement of pupil size and two-dimensional eye movements, Clin. Vis. Sci. 2 (1987) 131–142.

[23] K. Sung, D.J. Anderson, Analysis of two video eye tracking algorithms, Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. 13 (1991) 1945–1950.

[24] D.H. Ballard, C.M. Brown, Computer Vision, Prentice–Hall, Englewood Cliffs, New Jersey, 1982.

[25] A. Schwartz, Analytic Geometry and Calculus, Holt, Rinehart and Winston, New York 1960.

[26] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes in C: The Art of Scientific

Computing, Cambridge University Press, New York, 1992.

[27] W. Gander, G.H. Golub, R. Strebel, Least-squares fitting of circles and ellipse, BIT 34 (1994) 558–578.

[28] J. Merchant, R. Morrissette, J.L. Porterfield, Remote measurement of eye direction allowing subject motion over one cubic foot of space, IEEE Trans. Biomed. Eng. BME 21 (1974) 309–317.

[29] J.R. Charlier, J.C. Hache, New instrument for monitoring eye fixation and pupil size during the visual field examination, Med. Biol. Eng. Comput. 20 (1982) 23–28.

[30] H.F. Li, C. Eng, F.H.Y. Chan, P.W.F. Poon, W.F. Yan, W.Y. Wong, J.C. Huang, Real-time on-line pattern recognition of eye position and movement, IEE Proc. 132 (1985) 293–308.