Accurately Measuring the Size of the Pupil of the Eye

Xiang Lin, Gisela Klette, Reinhard Klette
CITR & Computer Science Department,
University of Auckland, Auckland, New Zealand

Jennifer Craig, and Simon Dean
Faculty of Medical and Health Sciences,
University of Auckland, Auckland, New Zealand

Abstract

A new method has been developed for accurate shape analysis of the elliptical contour of a human's pupil in images captured by a special infrared CCD video camera. The method is an effective combination of edge detection algorithms, a labelling method and a least squares fitting technique. The method is robust (eyes of different human races) and allows to trace the random movements of the pupil. We also solved the problem of dealing with the Purkinje phenomenon (i.e. highlights in captured images), which was previously only avoided by adjusting light directions [8]. The parameters of a fitted ellipse (length of both axes, orientation and centroid) are provided for further analysis.

Keywords: pupil size measuring, ellipse fitting, Purkinje.

1 Introduction

The size of the human pupil is an important parameter in different studies in Vision and Ophthalmology as well as in other areas such as Psychology or Psychiatry. Especially in excimer laser refractive surgery, accurate measurement of the pupil size of the eye in darkened conditions is an important key to the successful operation performance. A special infra-red pupillometer (the Dean Pupillometer) has been developed by *J. Craig* and *S. Dean* at the University of Auckland to overcome the problem. The images seem to be clear and simple: the dark pupil with obvious size is the only object in the image (Fig. 1).

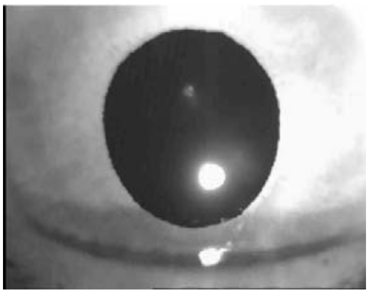

Figure 1: A captured image.

Unfortunately, there is still no article on how to measure accurately the pupil's size. Most pupil-related articles [3, 7, 9] are interested in the position of the pupil which reflects the orientation of eye-gaze. Some other articles involved in the size of pupils just treat contours of pupils as circles [1] and develop some simple algorithms [8] for roughly estimating the size of a circular region.

When we analyzed size and color of pupils in captured images, we actually found that it is not difficult to separate (approximately) a pupil from the background just by thresholding a whole image with a suitable threshold (Fig. 2). A more satisfying result can then be obtained by subsequent procedures of simple pixel labelling and region filling; see Fig. 3.

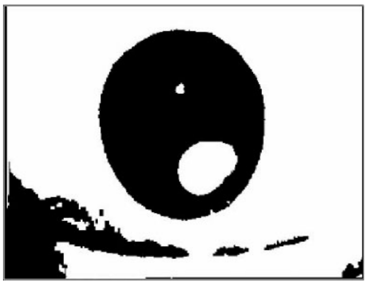

Figure 2: Image after simple thresholding.

The first problem is the automatic detection of the value of the initial threshold. A low threshold may cause the difficulty of separating a pupil from its surrounding area, especially when a part of background close to the pupil's contour is also quite dark. But a high threshold will also cause problems: it will reduce the size of the pupil and we may even lose completely part of the pupil if the the reflection area (Purkinje) is close to the pupil's edge (see Fig. 4). The selection of the initial threshold was so far based on experience. But a program needs to be designed to deal with all kinds of eyes: blue, gray,

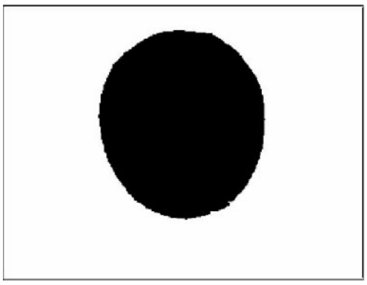

Figure 3: Image after applying labelling and filling procedures.

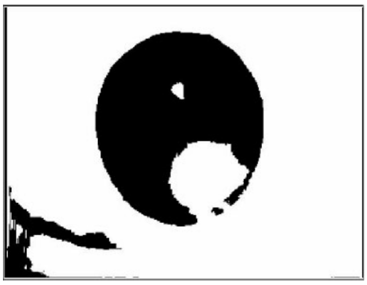

Figure 4: Image resulting from a high threshold.

brown or other colors, light or dark, with Purkinje or without. So it seems that we have to change thresholds by human interaction because experiments did not lead to any exact rule for value settings.

In this paper we employ edge detection (see, e.g., [5]) to solve this problem. The derived method is robust for eyes with different colors because it just depends on color changes in the images, not on the color itself. We first tested fast edge detection, and a method simple as the Sobel operator proved to be sufficient to generate "thick edges" from the original image which allow to identify the pupil's region by subsequent labelling. The edges of these regions are first reduced by applying a LoG operator to obtain "thin" contours, which drastically improves the execution speed of the final ellipse fitting program.

In general, there are two kinds of methods for calculating ellipse parameters from scatted points in an image, least square fitting [4] or Hough transform [6]. A Hough transform technique is normally applied to "noisy images", but with the drawback that it is slow especially when the number of Hough parameters is two or more. Under many variants of least-square ellipse fitting we decided for the widely used *direct least-square fitting of ellipses* [4]. To ensure real-time application of the least square fitting method for captured video sequences, we calculate "ideally thin" edges first. The whole process can be subdivided into five steps:

1. Rough (and simple) edge detection in the whole image.
2. Separation of edge regions of the pupil from edge regions of other image components.
3. More accurate edge detection for approximating the pupil's contour.
4. Thinning of the resulting edge.
5. Fitting of an ellipse to the thinned edge.

This article is structured in the order of these five steps.

2 Edge Region of a Pupil

The interesting segments in captured images are just the edge regions. The first step is to extract all edge regions. Our edge model includes that an edge region belonging to an individual object region should be connected and also separable from other edge regions.

For the first step, accuracy is not very important but speed is. The Sobel edge detector together with a relatively low and constant threshold proved to be sufficient to meet the requirement (see Fig. 5).

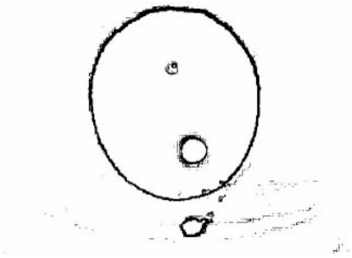

Figure 5: Sobel image (inverted image).

The simplest way to separate the edge of a pupil from other edge regions is to do region labelling in the whole edge image, and the largest region is chosen to be the pupil's edge. Unfortunately this takes too much time, especially for images with many small and separate edges. Our method is to find one *seed point* which is located in the edge region of the pupil, and then label all those edge points which are connected to the seed point.

We detect a seed point in the pupil's edge region by locating first a special pixel (D_x, D_y) in the pupil's area, which allows "to avoid" the reflection areas (Purkinje) in horizontal orientation, and then we search for a seed point within the pupils's edge region, starting from (D_x, D_y) in horizontal orientation.

The pupil is the only large-size black elliptical region in a captured image (see Fig. 6). Compared to the background, the darkness of a pupil is obvious. We divide the whole image into 8×8 blocks by experience; the center point of the darkest block must be in the pupil and we take this as (D_x, D_y). This pixel is "away" from Purkinje areas; otherwise, the block in which the point is located is by all experimental experience not the darkest one.

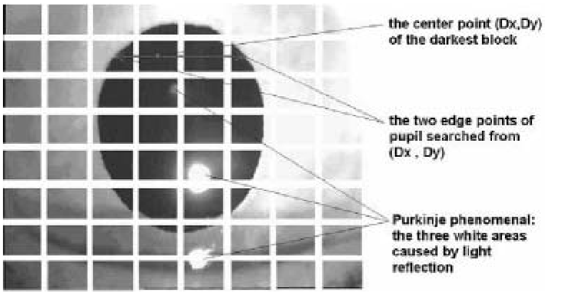

Figure 6: Center of the darkest block.

To get a seed point in the pupil's edge region, we analyze the horizontal line on which pixel (D_x, D_y) is located. By analyzing image values (Fig. 7) and their approximated first derivatives (Fig. 8), we evaluate the steepness of edges based on these diagrams, and choose the seed point in the middle of the steepest edge. (Note that there may be only one "column" in Fig. 8 if the pupil is close to the boundary of the image.)

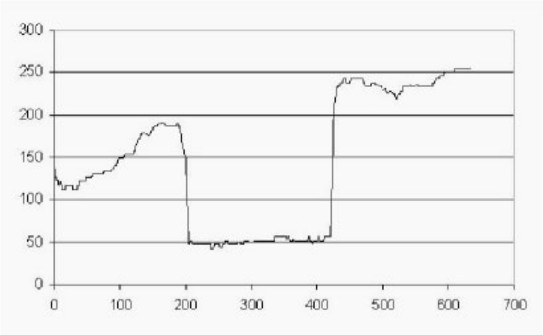

Figure 7: Curve of gray values.

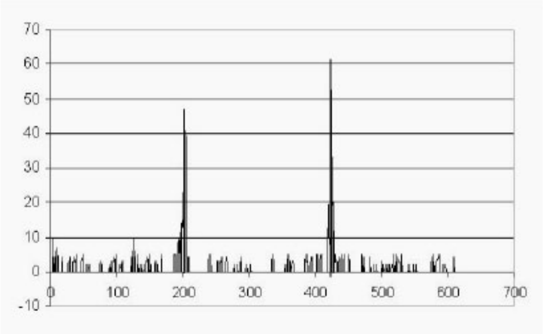

Figure 8: Curve of absolute values of first order derivatives.

The seed point allows finally a recursive fill of all the connected edge points (Fig. 9).

3 Accurate Edge Detection (Pupil Contour)

The contour of a pupil must be in the detected edge region. The well-known LoG operator combines a Gaussian filter and the Laplacian. The Laplacian allows a calculation of edges by estimating zero-crossings of second derivatives. We just calculate LoG-values (and zero-crossings) for pixels (using the original image data) in the detected edge region. The detected zero-crossings define a subregion of the previous edge region.

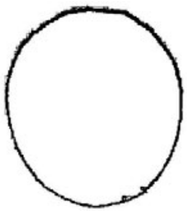

Figure 9: Rough edge region of pupil (inverted image).

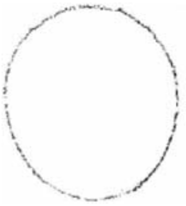

Figure 10: Edge pixels detected by LoG (inverted image.)

Our subsequent algorithm of direct ellipse fitting requires that detected pixels are evenly distributed on a simple digital curve. The more even the distribution is, the better the result will be. Here we employ a simple thinning algorithm to transform zero-crossings into such a simple curve.

First, the image is scanned line by line, and for every run of zero-crossings we have a leftmost and a rightmost point. We take the middle point of both, and these middle points form the final thin edge (Fig. 11). For balanced accuracy on the whole circle, we scan it in two orientations, horizontal and vertical (i.e. dividing the "circle" into four segments, and dealing with each segment separately).

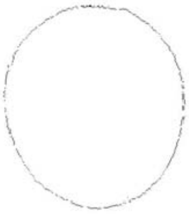

Figure 11: Edge after thinning (inverted image).

4 Direct Ellipse Fitting

Direct ellipse fitting is a robust, efficient, and easy method for fitting an ellipse to scattered data. "Direct" here means that all involved points are directly employed in calculating a least-square error, which is to find a set of parameters of an ellipse that minimize some distance measure between the data points and the ellipse. The direct ellipse fitting method incorporates the elliptical constraint into the normalization factor so that it can be solved naturally by a generalized eigensystem [4].

Direct ellipse fitting consists of two steps, first we calculate parameters of an ellipse in general quadratic form with a generalized eigensystem, then we translate the general quadratic form into a centered and oriented form.

Here we use a general conic to express the general quadratic form of an ellipse because an ellipse can be treated as a cutting section of a conic. So, the distance of a pixel (x,y) to the ellipse can be expressed as

$$F(\alpha,\chi) = \alpha \cdot \chi = ax^2 + bxy + cy^2 + dx + ey + f \qquad (1)$$

where $\alpha = [a\ b\ c\ d\ e\ f]^T$ and $\chi = [x^2\ xy\ y^2\ x\ y\ 1]^T$. The least square error is

$$\mathscr{D}(\alpha) = \sum_{i=1}^{n} F(\alpha,\chi)^2 \qquad (2)$$

for all pixels. To minimize it, we use a rank-deficient generalized eigenvalue system

$$D^T D\alpha = \lambda C\alpha\ , \qquad (3)$$

where $D = [\chi_1\ \chi_2\ \chi_3\ \cdots\ \chi_n]^T$, and a constraint matrix C,

$$\mathrm{C} = \begin{bmatrix} 0 & 0 & 2 & \cdots \\ 0 & -1 & 0 & \cdots \\ 2 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix},$$

which keeps $b^2 - 4ac < 0$. The whole process consists of the following steps:

1. Build matrix D with all the scattered pixels.
2. Create matrix S by $D^T D$.
3. Build matrix C.
4. Solve the generalized eigensystem.
5. Find the only negative eigenvalue.
6. Get the corresponding eigenvector as fitted parameters.

Now, we have the parameters of an ellipse in general quadratic form. The next step is to transform it into the centered and oriented form [2]. Here a small modification is made for the general quadratic form:

$$a_{11}x^2 + 2a_{12}xy + a_{22}y^2 + b_1x + b_2y + c = 0 \qquad (4)$$

If we assume that

$$\vec{Y} = \begin{bmatrix} x \\ y \end{bmatrix};\ A = \begin{bmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{bmatrix};\ \vec{B} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

and then assume

$$\vec{K} = \frac{1}{2(a_{12}^2 - a_{11}a_{22})} \begin{bmatrix} a_{22}b_1 - a_{12}b_2 \\ a_{11}b_2 - a_{12}b_1 \end{bmatrix},$$

we obtain

$$M = \frac{A}{\frac{\vec{B}^T A^{-1} \vec{B}}{4} - c}.$$

The form in equation (4) can be written in centered and oriented form

$$(\vec{Y} - \vec{K})^T M (\vec{Y} - \vec{K}) = 1\,, \qquad (5)$$

which can be eventually translated into the normal centered and oriented form

$$(\vec{Y} - \vec{K})^T RLR^T (\vec{Y} - \vec{K}) = 1\,, \qquad (6)$$

where

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}; L = \begin{bmatrix} 1/\mathrm{a}^2 & 0 \\ 0 & 1/\mathrm{b}^2 \end{bmatrix}.$$

Parameter a is the semi-major axis of the ellipse, and b is the semi-minor axis. θ is the orientation of the major axis. A final result is shown in Fig. 12.

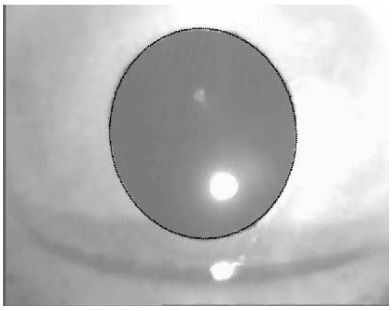

Figure 12: Ellipse fitted to the pupil's contour.

5 Required Subprocesses

The above method is based on the assumptions (i) that reflection areas (Purkinje) are not too close to the pupil's contour such that they create separated Sobel edge regions, and (ii) that the scattered pixels are (approximately) evenly distributed on a convex (elliptical) curve.

5.1 Overlapping Circles

For explanation purposes, we fake an original image by moving a reflection area into the edge region of a pupil (Fig. 13).

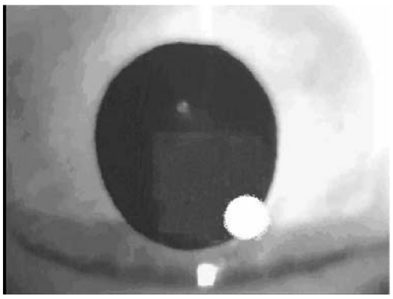

Figure 13: An image with a reflection area overlapping the pupil's edge region.

Figure 14 shows the Sobel edge region. Obviously, the big "circle" representing the pupil's edge cannot be separated from the small circle of the reflection area following the steps described above. The resulting thin edge introduces many incorrect edge points into the ellipse fitting step and eventually affects the result. The bigger the reflection area, the worse the result would be.

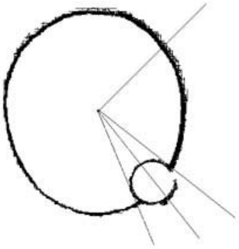

Figure 14: Sobel image for the assumed case (inverted image).

To reduce the effect caused by such (very unlikely) cases, a good solution is to erase a small circle from the Sobel edge image. The result will be improved because the fitting accuracy is better for missing points instead of additional incorrect points. For erasing we apply four steps:

1. Calculate the centroid of the whole region.
2. Radiate rays from the centroid and intersect them with the region.
3. If there is more than one run of intersection points, the ray must intersect with a small circle. All such intersection points are deleted.
4. Otherwise, the points are kept as edge points.

5.2 Scattered Points

In generalization of the previous method, we discuss the case as typically occurring in captured images: scattered points do not form "circles", there can be several "overlaps" with different edge fragments. The general procedure is as follows:

1. Calculate the centroid for the Sobel (or zero-crossing) edge regions.
2. Calculate the relative orientation of edge points to this centroid, for every point and save it as an integer value $(0,\ldots,359)$.
3. Calculate the distance between the points and the centroid for every point.
4. For each degree $0,\ldots,359$, check if there are any two points with the same degree whose distance is bigger than a predefined threshold. If it is, delete all points with this degree from the image. The predefined threshold must be bigger than the "thickness" of the thickest part of the "big circle".

A result is shown in Fig. 15.

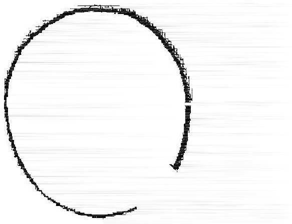

Figure 15: Sobel image after eliminating the small circle (inverted image).

See Fig. 16 and Fig. 17 for a visual comparison of a difference between ellipses calculated with and without prior data cleaning (in the Sobel image), respectively.

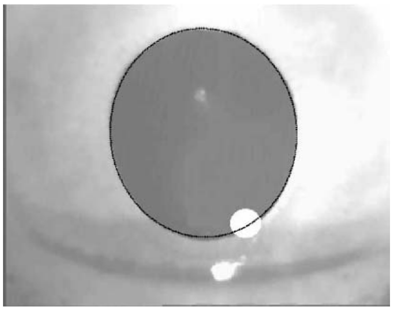

Figure 16: Fitted ellipse without prior data cleaning.

6 Conclusions

A Dean Pupillometer video clip (avi file in mpeg4 format) is typically about 6-seconds. The task is to measure the minimum, maximum and mean values of pupil

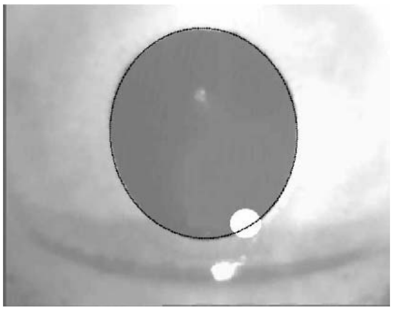

Figure 17: Fitted ellipse for the same image but after data cleaning.

sizes which are required to be represented as the radii on both axes of fitted ellipses. Figure 18 illustrates calculated axes, and Table 6 shows values in pixels (for one sequence).

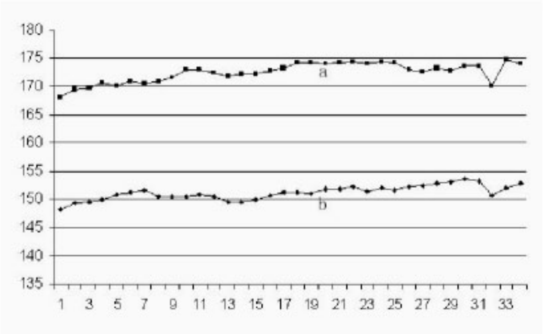

Figure 18: Results of radii in 36 images of a video clip.

Table 1: minimum, maximum and mean values

	value a	value b	average
average	172.3	151.1	161.7
maximum	174.7	153.6	164.2
minimum	167.2	148.2	157.7

The developed method is able to measure the elliptical parameters of human pupils in real time (analyzing 6 images per second on a Pentium III 800MHZ PC) by processing a sequence of images captured from 5 or 6 second video clips. The accuracy of obtained results is at subpixel level, which is an improvement compared to other published approaches.

The robustness of the method is also verified by a group of video clips taken from persons with different colors of eyes. The method does not imply any limitations for the color of the eye, the size range of the pupil or the illumination conditions. Purkinje phenomenons have been taken into consideration and attention has been paid even to rarely occurring cases which could affect the estimated size of pupils. Basically the only requirement is that the depicted pupils do not cross the image boundary. For such cases we would need an algorithm for ellipse fitting which is able to cope with situations where only a segment of the ellipse is provided. Nevertheless, this is not a serious limitation in practice because such crossings can be avoided by the Dean pupillometer.

Parameter calculations in this paper are in pixel units. However, we also have a calibration method for transforming pixel units into physical units.

References

[1] B. Chou and B. Wachler. The role of pupil size in refractive surgery. In A. Agarwal and S. Agarwal, editors, *Textbook of Ophthalmology, Volume 1.* Jaypee Brothers, India, 2001.

[2] D. Eberly. Information about ellipses. http:// www.magic-software.com/Documentation/ InformationAboutEllipses.pdf. (accessed: 9 Dec. 2002).

[3] Y. Ebisawa. Unconstrained pupil detection technique using two light sources and the image difference method. In *Visualization and Intelligent Design in Engineering and Architecture II*, pages 79-89. Computational Mechanics Publications, Southampton Boston, 1995.

[4] A. Fitzgibbon, M. Pilu, and R.B. Fisher. Direct least square fitting of ellipses. *IEEE Trans. Pattern Analysis Machine Intelligence*, **21**:476–480, 1999.

[5] R. Klette and P. Zamperoni. *Handbook of Image Processing Operators.* Wiley, Chichester, 1996.

[6] S. Inverso. Ellipse detection using randomized Hough transform. http://www.cs.rit.edu/ sai6189 /vision/final/EllipseDetection.pdf (accessed: 11 Dec. 2002).

[7] C. Lankford. Effective eye-gaze input into windows. In *Proc. Eye Tracking Research and Applications Symposium*, pages 23-27. ACM, 2000.

[8] I. Miro, N. Lopez-Gil, and P. Artal. Pupilmeter and tracking system based in a fast image processing algorithm. *Proc. SPIE,* **3591**:63–70, 1999.

[9] N.M. Taylor, R.H. Eikelboom, P.P.v. Sarloos, and P.G. Reid. Determining the accuracy of an eye tracking system for laser refractive surgery. *J. Refractive Surgery*, **16**:643–646, 2000.