Efficient and Robust Pupil Size and Blink Estimation from Near-Field Video Sequences for Human–Machine Interaction

Siyuan Chen, *Student Member, IEEE,* and Julien Epps, *Member, IEEE*

***Abstract*—Monitoring pupil and blink dynamics has applications in cognitive load measurement during human–machine interaction. However, accurate, efficient, and robust pupil size and blink estimation pose significant challenges to the efficacy of real-time applications due to the variability of eye images, hence to date, require manual intervention for fine tuning of parameters. In this paper, a novel self-tuning threshold method, which is applicable to any infrared-illuminated eye images without a tuning parameter, is proposed for segmenting the pupil from the background images recorded by a low cost webcam placed near the eye. A convex hull and a dual-ellipse fitting method are also proposed to select pupil boundary points and to detect the eyelid occlusion state. Experimental results on a realistic video dataset show that the measurement accuracy using the proposed methods is higher than that of widely used manually tuned parameter methods or fixed parameter methods. Importantly, it demonstrates convenience and robustness for an accurate and fast estimate of eye activity in the presence of variations due to different users, task types, load, and environments. Cognitive load measurement in human–machine interaction can benefit from this computationally efficient implementation without requiring a threshold calibration beforehand. Thus, one can envisage a mini IR camera embedded in a lightweight glasses frame, like Google Glass, for convenient applications of real-time adaptive aiding and task management in the future.**

***Index Terms*—Biometrics, cognitive informatics, human factors, image segmentation, pervasive computing.**

I. Introduction

EYE TRACKING has been a research area of interest for decades. In most eye tracking systems (for a detailed review see [3]), the inherent objective is to determine the center of the pupil for gaze direction, e.g., in human–computer interaction [1]–[4]. In this case, the precision of pupil size and blink is not critical, since four points that satisfy the crossed chord theorem can determine the center of the pupil [5], [6], and a blink, causing loss of the point of gaze, only occurs in a few hundred milliseconds. By contrast, for monitoring eye activities for cognitive load estimation [7], precise measurement of pupil size and blink duration in real time is required in order to reflect the instantaneous mental activity, so that timely feedback or assistance from machines can be provided to improve human–machine interaction (e.g., adaptive automation [44]). Meanwhile, efficient and robust measurement is important for applying it to a wide range of scenarios.

By definition, cognitive load occurs when task demands are imposed on the human cognitive system due to limited mental resources [33]. Measurement of cognitive load (different name might be used) can be traced back to the late 1930s and has roots in the field of human–machine interaction. With the aim of improving task performance and maintaining health and safety [34], physiological signals, such as electroencephalogram and eye movement, were employed and created high expectations in the so called biocybernetic projects at that time [34]. In particular, after blink, pupillary response, and eye movement were discovered to be associated with cognition in around the 1920s [35] and the 1960s [36], [39]–[41], eye-based cognitive load estimation systems were motivated by research into adaptive interfaces [9], monitoring task characteristics [8], managing interruption time [37], and differentiating expert and novice acts [38], to name a few. Especially, real-time processing of the captured eye activity can allow machines to provide an adaptive task manager style measure of load on human processing capacity, existing measures for which are currently performed manually and are, hence, completely unfeasible for a myriad of computing applications.

To develop such an eye-based real time cognitive load measurement system, the technical barrier of data acquisition has not been completely resolved if current commercial eye trackers are to be employed for this purpose. To acquire eye activity, unlike for commercial eye trackers, calibration and gaze direction computation are unnecessary, but the output pupil size and blink duration must be precise and ideally obtained from real-time image processing. However, this application has not drawn much attention from the image processing research community as cognitive load researchers have, until now, usually relied on commercial eye trackers to measure pupil size and blink, which is often a by-product output. Not

Manuscript received July 30, 2013; revised November 19, 2013 and February 6, 2014; accepted February 10, 2014. Date of publication March 21, 2014; date of current version November 13, 2014. This work was partly supported by NICTA which is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program. This paper was recommended by Associate Editor B. Chaib-draa.

The authors are with the School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, NSW 2052, Australia, and ATP Research Laboratory, National ICT Australia, Sydney, NSW 2015, Australia (e-mail: siyuan.chen@student.unsw.edu.au; j.epps@unsw.edu.au).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2014.2306916

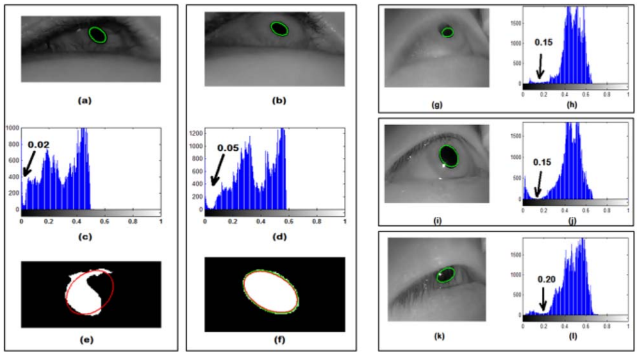


Fig. 1. Eye images with different variations and their histograms. (a), (b), (g), (i), and (k) Green curves in gray images are the fitted ellipses after binary segmentation with the threshold selected by using the first dip point of their (c), (d), (h), (j), and (l) corresponding intensity histograms. (a) and (b) Eye images from the same person in different days and pupil size can be estimated using 0.02 and 0.05 as binary thresholds, respectively. If their thresholds are set to be 0.05 or 0.02 (without tuning thresholds), the resulting pupil size will be larger (by around 0.19 mm) or smaller (by around 0.14 mm) than the true size. Estimated pupil size can also be smaller than true pupil size in (g) due to reduced contrast resulted by motion blur, in (i) due to corneal reflection near pupil boundary, and in (k) due to eyelid occlusion, if we use the first dip point of the histogram to estimate the threshold.

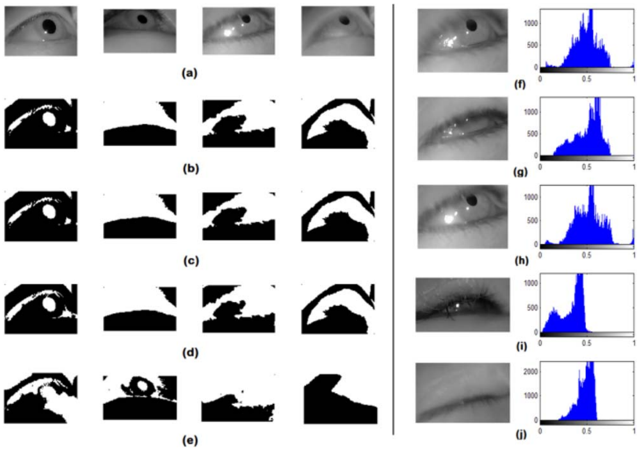


Fig. 2. (a)–(e) Some traditional image segmentation techniques are not applicable to all eye images with different variations. (a) Examples of four eye images recorded from different people and on different days. Binary thresholding results using the (b) Otsu method [17], (c) iterative threshold method [18], (d) minimum cross entropy method [19], and (e) robust automatic threshold selection method [20], [21]. (f)–(h) Three consecutive eye image frames and their corresponding histograms, showing blink can be as short as only one frame (33 ms). (i) One blinking frame where the eyelids are open and its histogram. This blink lasts five frames. (j) One blinking frame where the eye is fully closed and its histogram. This blink lasts nine frames. This shows that using the distance of two eyelids might not be good enough to determine whether it is a blinking frame.

only is the measurement accuracy unknown (instead the gaze resolution is often given), but also it requires recalibration when environments and users change.

To obtain precise pupil size and blink endpoints, a near-field eye monitoring system is more desirable than a far-field system, and also ideal for a mobile computing usage [10]. Embedding an IR mini camera in eyewear and pointing it toward the eye is a simple step. However, the traditional image processing techniques suffer some problems, discussed further in Sections II and III, making accurate measurements nontrivial.

This paper contributes in two aspects: 1) It investigates the challenges of obtaining accurate and robust measurement of pupil size and blink from near-field videos with large variations due to different users, tasks and environments, with supporting evidence from a realistic video dataset and 2) it proposes novel methods to estimate pupil size and blink endpoint without tuning parameters for managing these variations, thus it alleviates the inconvenience of calibrations when users are changed, tasks are different, and cameras are repositioned near the eye, and enhances the autonomy of the system.

II. Problem Statement

The first challenge is that pupil shape is not constant during tasks, which is the main difference from iris detection, as the iris does not change shape over time. This renders some shape-based methods, e.g., deformable eye model or voting scheme [23], which rely on prior models and similarity measures, nonrobust and inefficient, as a prior model hardly represents all pupil shapes. Pupil shape can change dramatically for the following reasons: 1) The camera and infrared (IR) source are often placed below the eye and point upward in order not to block the line of sight. This causes an off-axis and noncircular view of the pupil in the image; 2) the pupil shape can be an ellipse, half or partial ellipse depending on the magnitude of a blink. During a full blink, the pupil completely disappears, making it even worse; and 3) at any time, the pupil can dilate or constrict by as much as a 10% change (Fig. 3(a)) in its size between frames (recorded at 30 frames/s) due to fast pupil rotation. Other factors such as light reflex, cognitive load, or arousal can also make pupil size change rapidly [8]. Therefore, the approaches that are effective for constant and circular iris segmentation, such as the integrodifferential operator [11] or edge detection followed by Hough transforms [12], cannot be directly used to locate the pupil and estimate its size. To deal with the noncircular boundary problem, ellipse fitting is one of the feasible approaches given valid points on the pupil boundary, and has often been used to estimate pupil size (the length of the major axis) [1], [5], [13], [14]. However, its accuracy depends on the quality of pupil segmentation from the background, as shown in Fig. 1(a)–(f).

The second challenge is due to noise from the eyelid, eyelash, and illumination variations that always accompany and influence pupil segmentation that makes parameter selection important but difficult for video datasets with large variations. Although an IR light source can produce a distinct contrast between the pupil and background to some extent, with a close view of an eye, problems in segmentation can become more severe than for the eye image in the far field. These problems can be due to eyelid occlusion and shade (Fig. 1(a)), eyelash occlusion (Fig. 1(k)), the corneal reflection on pupil boundary (Fig. 1(i)), blur due to defocusing, or eyelid motion (Fig. 1(g)).

To lower the cost of manual threshold selection in order to deal with eye image variations, an automatic or adaptive threshold method is much desired. However, some thresholding techniques [15] that work efficiently, for example, for images with bimodal histograms, might not yield ideal pupil segmentation, as shown in Fig. 2(b)–(e).

Meanwhile, locally adaptive thresholding methods cannot produce better pupil segmentation, e.g., Fig. 2(e). Despite the decision for an appropriate window size, pixels in each local window might not have the desired intensity distribution for robust binary segmentation. For example, local windows that contain only skin and eyelash pixels can yield unwanted binarization. Local windows that contain iris, eyelid shade, and pupil pixels have similar issues to global thresholding. Ji and Yang [16] demonstrated that if the intensity distribution is unimodal in such windows not containing pupil pixels and is bimodal in windows that include pupil pixels, the locally adaptive threshold is very robust.

However, in their study, off-axis and on-axis IR light sources were switched on and off in turn to produce dark and bright pupil images in sequence. Pupil detection was performed on the difference image of the two. As the IR light sources need to be synchronized with camera recordings, the hardware is more complicated than an IR diode merely soldered on a web camera board for recording. Meanwhile, large and fast pupil motion that occurs between the dark and bight pupil images in sequence can cause troubles, thus extra sensors or methods are often required for compensating the hardware [3]. Hence, this approach was not adopted in our paper.

III. Related Work

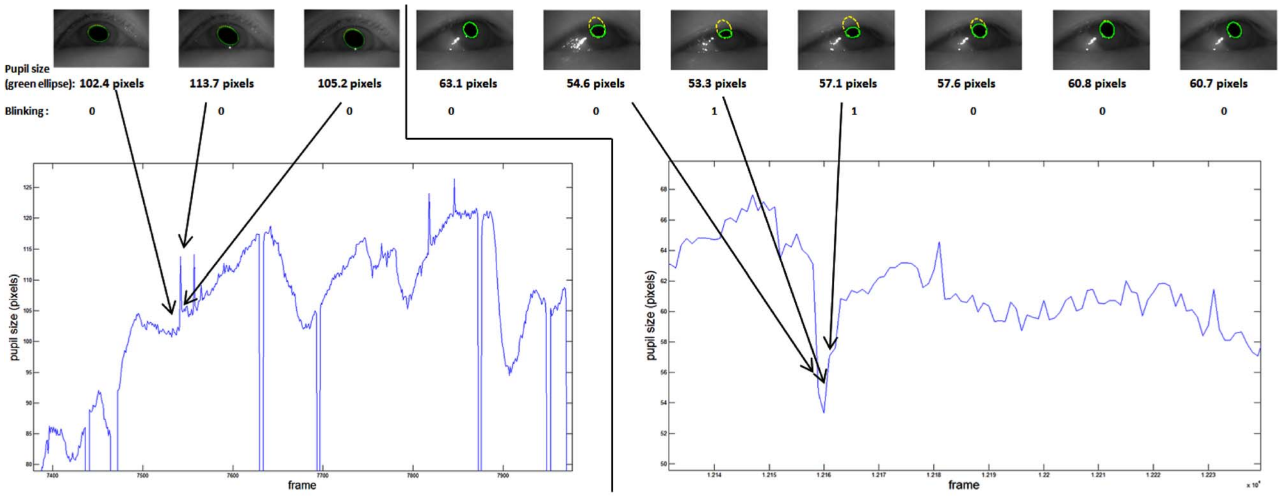
As Fig. 1(e) shows, inappropriate binary thresholding can result in irregular pupil shape. The irregularity can be due to eyelid shade, eyelash occlusion and corneal reflection on the pupil boundary. This makes ellipse fitting very difficult if the pupil boundary points, which do not represent the true pupil shape, are used for fitting. Zhu *et al.* [13] proposed a curvature algorithm to find the appropriate points that are on the true pupil boundary for ellipse fitting. In their method, pupil blobs were firstly identified in a binary image after a threshold operation and the elimination of spurious blobs. A curvature value was then computed for each pupil boundary point. Any point on the boundary occluded produced a peak in curvature value. Therefore, only the boundary points within the upper and lower curvature thresholds participated in the ellipse fitting. The curvature algorithm intends to fix the problems resulted by pupil segmentation given a fixed binary threshold. Although it can reduce the errors to some extent, the curvature threshold depends on the pupil radius. That is, without a known pupil radius, it is hard to determine appropriate curvature thresholds. Thus, it does not meet our requirement where the pupil size has not been measured and often changes rapidly.

Similar papers focused on pupil boundary point filtering rather than accurate pupil segmentation that used different methods before ellipse fitting. For example, iterative ellipse fitting for the boundary pixels after a double threshold for pupil segmentation [5], and the random sample consensus (RANSAC) paradigm on detected edge points to remove outliers for a best-fit ellipse [1]. In these methods, the number of iterations or convergence threshold needs to be empirically set, and it might not guarantee that each image can converge within a certain number of iterations or to that convergence threshold in any context. Meanwhile, their robustness to the noise from eyelid and eyelash shade has not been demonstrated.

It is easy to observe that when the eye is open, the intensity of eye images can be roughly divided into four levels, e.g., Fig. 2(a). The skin and sclera have the brightest intensity level, followed by the iris. The eyelash and eyelid shade have the third level whereas the pupil has the darkest level with IR illumination, and only the darkest level is of interest. Thus, De Santis and Iacoviello [14] proposed a region-based optimal segmentation procedure to estimate pupil parameters. They used the framework of the level set formulation from the Mumford and Shah model to design their energy function. Four-level segmentation was obtained in a resized image, and the steady-state level set was the optimal segmentation to use. Estimated pupil size was compared with that using a threshold procedure by the first dip point of the intensity histogram [14]. With the level set method, the measured pupil size was found to be less affected by noise and was closer to the true pupil size. However, this method is sensitive to the choice of a parameter that requires a higher value setting for low contrast images. Meanwhile, the quality of segmentation and the expense of computation depend on the convergence criteria in the iteration process. Thus, it is less efficient for real-time processing with vast variations in eye contrast, pupil position, and appearance in video sequences.

A variety of methods in the literature, although achieving encouraging performance, have limitations related either to difficulty determining parameters or computational complexity due to exhaustive searches over a large parameter space [22], [23]. For example, the Starburst algorithm for the locations of the pupil and corneal reflection [1]; an iris tracker combining particle filter with the EM based active contour [23]; mean-shift based segmentation [24], [25], and graph-based segmentation [25], [26]. Other methods can be drawn from supervised learning. Although high performance can be achieved, e.g. for iris detection [22], they have an unsolved shortcoming in the high cost of building a training set. Therefore, they are not suitable for processing video sequences with large variability and concerns about manual cost. Furthermore, the existence of blink was often ignored or treated separately in these related papers.

Accurate blink endpoints are desirable in order to estimate blink duration and to interpolate the pupil size during blinks. With commercial eye trackers, the blinks are often recognized based on obtained pupil size, that is, when the pupil size is zero (fully closed) (defined as the iris disappearing in [23]) or below a threshold (partially closed) [27]. These assume that an eye tracker does not lose the eye [23], and blink is the only cause of significantly reduced pupil size. In practice, studies often select consecutive frames when pupil size is zero for a period of time, e.g., at least 83.3 ms in [9] (within a normal blink duration), as blinking frames. However, during a task, a blink can be as short as 33 ms, as shown in Fig. 2(f)–(h). Meanwhile, pupil size can be reduced rapidly when the background is switched from dark to bright or a cognitive task is completed. Therefore blink detection based on empirical



(a) (b)

Fig. 3. (a) Example of three consecutive video frames where a fast pupil rotation caused a 10% increment in pupil size followed by an 8% constriction, corresponding to a spike in the pupil size plot. For cognitive load measurement, the spikes were removed by a median filter after an interpolation during blinks. (b) Example of seven consecutive video frames where the eyes were not fully closed during a blink, causing a valley in the pupil size plot. Empirically setting a threshold of 20% or 50% of the previous maximum pupil size or degree of eyelid closure for blink poorly identifies this blink, whereas the proposed dual-ellipse scheme successfully identified it. These demonstrate that during a task, eye activity has more variation in pupil size and blink than that during a nontask circumstance, and using empirical setting based on normal conditions might not be reliable.

criteria, such as duration of zero pupil size and percentage of previous and subsequent pupil size, is unreliable, as shown in Fig. 3.

For some model-based methods, blinks need to be detected before applying them to open eye images to satisfy the model assumption, e.g., the eye template model [28]. A proposed method is to detect closed or open eye states firstly based on iris intensity and edge [28]. Active appearance modeling technique is another approach to detect blink by modeling eyelid motion, where the distance of upper and lower eyelids is compared with a predefined threshold for blink [29]. However, these methods are not applicable to every case, as Fig. 2(g) and (i) show, the eyelid can be opened and the iris is still visible when the pupil is blocked and the vision is fully suppressed. With a remote camera, the blink can be detected by computing correlation scores to an open-eye and a closed-eye template [4] or by determining whether there is a significant difference between successive images in the two eye regions [30]. These methods do not require pupil information, but similarly, the efficiency and robustness of these methods often depend on the choice of threshold values, and also they are more suitable for remote eye detection with a lower resolution in the eye regions.

Obtaining the information on pupil size, blink, and gaze (the center of the pupil; it can be relative to the head) in an eye-tracker system can enable interesting new applications in human–machine interaction. A recent paper [8] proposed three types of eye activity features which automatically and continuously detect three task characteristics in real time–task transition, perceptual load, and cognitive load, so that the machines can monitor and analyze dynamic human task states and intensities to offer timely feedback for a more effective human–computer partnership.

IV. Method

For the eye activity acquisition device, we dismantled two off-the-shelf IR webcams (30 frames/s) and soldered two IR LEDs (160 mW) in each camera board. The IR light wavelength is 940 nm, and the filter of visible light above the lens was replaced by an IR filter [42]. Then, we mounted the IR webcams on a pair of safety glasses. The webcams were placed pointing upward toward the eye to capture near-field eye images. The distance was adjusted to be around 5–6 cm, so that the total exposure of irradiance was below the recommended safety level of 10 mW/cm^2 [43].

We also developed a self-tuning threshold algorithm. The principle of this algorithm is similar to methods, such as that by Otsu [17] and minimum cross entropy [19], where the optimum binary threshold was determined by minimizing a criterion via an iterative search. In the proposed method, a gradient function of pupil coordinates and area is developed rather than a gradient function of intensity [20] or a function based on its histogram [17], [19], where global spatial information is often lost. A convex hull was then used to select pupil boundary points for ellipse fitting. The occlusion state was detected when two fitted ellipses, using the whole pupil boundary and the lower half of pupil boundary respectively, were different. Therefore, blink detection is implemented at the same time and based on current pupil size estimation. Details are described below.

A. Task Settings and Data Collection

This video dataset was collected for predicting three task characteristics from eye activity. Twenty two participants performed three continuously switched tasks on a computer over two different days. Each task was presented in different backgrounds, white, gray, and black. The three types of tasks were designed in three perceptual load levels. Within each type, the tasks were in five induced cognitive load levels. Participants wore head-mounted IR webcams, and the recordings were made for each of the sessions at 30 frames/s, with duration of 10 to 25 min, as described in [8]. Therefore, the dataset of eye images contained all variations due to task type and difficulty, different illumination, eye shape, distance between webcam to the eye, camera angle, and ethnicity. They reflect most variations in reality and the inclusion of all of these types of variations makes the database quite a realistic one.

B. Blob Analysis to Locate Pupil Blob

Each eye image was firstly resized (the size was manually determined in the first video frame and used afterward), to yield an image I with width W_{I} and height H_{I}. As the width represented the eye length in the image, the actual estimated pupil size can be easily converted by the ratio of W_{I} to the physical length of the eye. It was then converted to a gray-level image, and was inverted (I_{G}) to make the object of the pupil as the brightest pixels, before median filtering (I_{GF}) to remove noise and a contrast adjustment by linear mapping (I_{GFC}). In the binary image produced by thresholding t_{i} (i is the threshold index) on I_{GFC}, each blob, with area A (number of white pixels), was located by a bounding box, with four corner positions (H_{up}, H_{down}, V_{left}, V_{right}), as a pupil candidate. Only one blob with bounding box was selected as the true pupil blob in the binary image, as explained in Algorithm 1.

To identify the true pupil blob, blobs that did not meet the following rules (in order) were eliminated: 1) $A_{\mathrm{i}} < A_{(\mathrm{i}-1)}$ (area should be increasing or identical when threshold t decreased (from $t_{\mathrm{i}-1}$ to t_{i})); 2) $H_{\mathrm{right}} \leq (W\text{-}10)$ or $H_{\mathrm{left}} \geq 10$ (positions of pupil blob cannot be at the edge of the eye image); 3) maximum pixel value in the bounding box in I_{GF} is smaller than the maximum pixel value from all bounding boxes in I_{GF} (the true pupil blob should contain the highest pixel value due to dark pupil effect); 4) ratio of blob area to bounding box area

$$\frac{A}{(H_{down} - H_{up}) * (H_{right} - H_{left})} \leq 0.5$$

5) ratio of width to height

$$\frac{H_{right} - H_{left}}{H_{down} - H_{up}} \notin [0.25, 4].$$

Some constraints were set empirically and often resulted in one or several pupil candidates. 6) If the current threshold is the first highest threshold that produces blobs, then the blob that has the maximum area was selected as the true pupil blob; otherwise the blob whose bounding box resulting from t_{i} was the closest to that of the identified true pupil blob resulting from $t_{\mathrm{i}-1}$, was the true pupil blob. The metric used to indicate the distance is ℓ_1 norm using the top-left corner of the bounding boxes. If no blob was selected, typically when the pupil was fully occluded, the position of the bounding box was $(-1, -1, -1, -1)$.

Among the five rules, rule 1 and 6 are the most important rules to select correct blobs and effectively formed an association rule. That is, the association of the blobs from one threshold $i-1$ to the next threshold i is used in order to consider whether it is the same blob and thus, to contemplate the evolution of its area and positions.

Algorithm 1 Blob Analysis Using Association Rules

```
Input: $I_{\mathrm{GFC}}$
Output: one bounding box location (bbox)
begin Initialize bbox to be $[-1, -1, -1, -1]$
$t = 1$:-$s$:0 ($s$ is the step of the threshold)
for $i = 1$:length($t$)
  threshold $I_{\mathrm{GFC}}$ with $t_i$ to obtain binary $I_{\mathrm{GFCB}}$
  If there are $n$ blobs in $I_{\mathrm{GFCB}}$, calculate blob area $A_n$
   and bounding box positions box $_n$
    If $i \neq 1$, select the blobs whose area is larger than $A_{i-1}$
    (Rule 1), end
    If more than 1 blob exists, select the blobs whose
    box $_n$ is not at the edge (Rule 2), end
    If more than 1 blob exists, select the blobs whose
     box $_n$ contains the maximum pixel value in $I_{\mathrm{GF}}$
     (Rule 3), end
    If more than 1 blob exists, select the blobs whose
     area ratio and length ratio are within the range in
     Rule 4 and 5, end
    If $i = 1$, select the blob whose blob area is the largest
     one
    else if bbox$_{i-1} \neq [-1, -1, -1, -1]$, select the blob
     whose blob area is the largest one
    else select the blobs whose box $_n$ is the closest to
     bbox$_{i-1}$ (Rule 6)), end
    bbox$_i$ = box $_n$
  end
  End
```

As the threshold is always swept from 1 (initially) toward 0, blobs are always being merged when the threshold decreases, thus rule 1 is an intuitive criterion for filtering blobs associated with previous thresholds. If the first threshold is too high to produce any blob, the second threshold will, and so on. Since the true pupil blob contains the highest pixel values (an important assumption forming rule 3), the first few highest thresholds can guarantee to produce blobs that are part of the true pupil blob. Therefore previous blob location was used to filter blobs resulting from lower thresholds, forming rule 6. When the threshold decreases, more pixels from the true pupil blob merge to change its shape, being more close to the true pupil blob. When the threshold is low, irrelevant pixels are incorporated, and its bounding box either expands to the edge of the image, violating rule 2, or is too slim, violating rule 5, or the ratio of blob area to bounding box area is too low, violating rule 4.

C. Self-Tuning Threshold Algorithm to Obtain Optimal Threshold for Segmentation

$E(t)$ is the proposed gradient function of the binary threshold $t \in [0, 1]$, with a step of $s \in (0, 1)$. The idea is that within a certain range of t, the area and positions of the segmented pupil have little variation near an optimum threshold. By decreasing t beyond the optimum value, the resulting binary image will

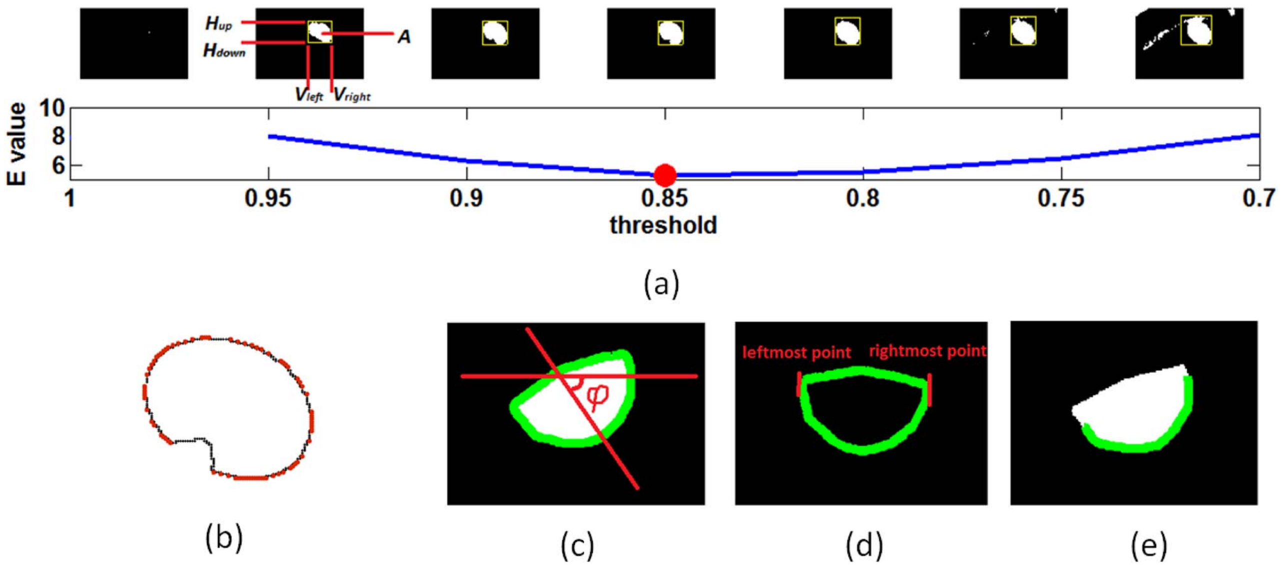


Fig. 4. (a) Example of an open eye binary image for different threshold values (top) with a step of 0.05 and the corresponding $E(t)$ values plotted against t (bottom). The object (pupil) is in white pixels. The optimum threshold is denoted by a red dot. (b) Example of convex hull for a selection of pupil boundary points. The dark points, P, are the pupil boundary points and the red points are the convex hull of P. The points on the notch are removed to minimize the distortion for ellipse fitting. (c) Example of using all the points on the pupil boundary. (d) Rotated pupil boundary of (c). (e) Lower half of pupil boundary for ellipse fitting. The pupil boundary points are denoted by the green curve. (c) and (e) Binary images of Fig. 1(k).

have the pupil connected with pixels from the eyelids, causing the area and locations to expand. By increasing t beyond the optimum value, the area and positions of the segmented pupil will keep reducing and shrinking until the pupil diminishes. s acts as the resolution of the threshold, the smaller the better but with higher computational expense.

By sweeping t from 1 to 0 with step s, and iteratively performing blob analysis at t, we obtain vectors of $A(t)$, $H_{\text{up}}(t)$, $H_{\text{down}}(t)$, $V_{\text{left}}(t)$, $V_{\text{right}}(t)$. For each value of t

$$E(t) = |\nabla A| + |\nabla H_{up}| + |\nabla H_{down}| + |\nabla V_{left}| + |\nabla V_{right}| \quad (1)$$

$$t_{opt} = \arg\min_{t}(E(t)) \quad (2)$$

where ∇ is the gradient operator, using a central difference scheme with -1 padding. E is filled by positive infinity for t when there is no pupil blob (i.e., the position of the bounding box is -1), hence excluded from optimal threshold selection. t_{opt} is the optimal threshold if there is a pupil blob at t_{opt}, otherwise it returns empty.

If no optimal threshold was found, the current state of the pupil was denoted as fully covered, otherwise, the optimal threshold was used to segment the pupil blob from the background in I_{GFC}. The nonpupil area was masked so that only one component was in the resulting binary image. The optimal threshold can ensure that the segmented pupil blob is isolated from the eyelids to the maximum extent. It can also include the pixels inside the pupil blob with lower intensity due to uneven illumination, pupil shape, and size variations, as per the problem shown in Fig. 1(e) and (f). An example of the self-tuning threshold is illustrated in Fig. 4(a).

D. Convex Hull to Select Pupil Boundary Points

A convex hull method was used to solve problem in Fig. 1(i), where a notch occurs in the pupil blob due to the corneal reflection near the pupil boundary. By definition, the convex hull of a set of points, P, is the smallest convex set enclosing all the points of P. Fig. 4(b) depicts the selection by applying a convex hull function to all boundary points.

E. Ellipse fitting to estimate pupil size and detect blink

We used a least squares approach to estimate the ellipse parameters for the best fit to the pupil boundary

$$ax^2 + bxy + cy^2 + dx + ey + f = 0. \quad (3)$$

We define $\boldsymbol{G}$ as a matrix of $[\boldsymbol{X}^2, \boldsymbol{XY}, \boldsymbol{Y}^2, \boldsymbol{X}, \boldsymbol{Y}]$. $\boldsymbol{X}$ and $\boldsymbol{Y}$ are the column vectors of single measurements of pupil boundary points (x, y). $\boldsymbol{L} = [a, b, c, d, e]^T$ is the parameter vector to be estimated. The parameters have been normalized by $(-f)$ in (1), so that $f = -1$. The least squares approach yields an estimate for $\boldsymbol{L}$, the orientation of the ellipse, φ, the center coordinates, (x_0, y_0), and the length of the subaxes a' and b', of the ellipse

$$\widehat{L} = [G^T G]^{-1} \sum G \quad (4)$$

$$\varphi = \frac{\arctan(\frac{b}{c-a})}{2} \quad (5)$$

$$x_0 = \frac{-d}{2a}; y_0 = \frac{-e}{2c} \quad (6)$$

$$a' = \sqrt{|\frac{f'}{a}|}; b' = \sqrt{|\frac{f'}{c}|} \quad (7)$$

where

$$f' = -f + \frac{d^2}{4a} + \frac{e^2}{4c}.$$

The first ellipse was fitted with all pupil boundary points, as shown in Fig. 4(c), to obtain the major axis $a_1 = \max(a', b')$,

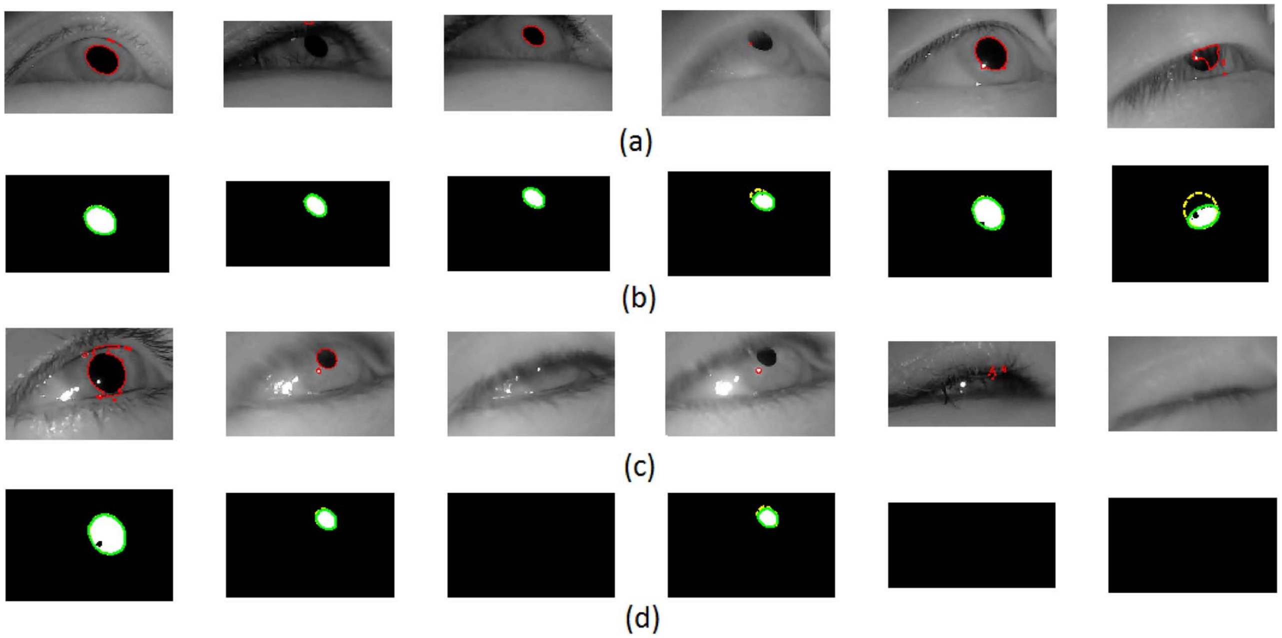


Fig. 5. Segmentation results. (a) and (c) Examples of eye images from different people and in different days. The red curves overlaid are the results of the zero-level set using the level set method [32] with fixed parameters and after 210 iterations. (b) and (d) Segmentation and ellipse fitting results using the proposed methods in their corresponding binary images. A blank binary image indicates that the current frame is a blinking frame. When the eye is open, two ellipses were fitted, denoted by green and yellow curves (if the two ellipses are identical, the yellow curves are not visible). The degree of pupil occlusion can be indicated by the extent of the difference between the two ellipses. These demonstrate that the level set method is able to estimate pupil size when the contrast is strong and there is little noise around the pupil, but it is not able to deal with all images with different variations without adjusting parameters, whereas the proposed methods can.

minor axis $b_1 = \min(a', b')$, and orientation φ_1. The second ellipse was fitted with points from the lower half of the pupil boundary, as shown in Fig. 4(e). To obtain these points, all boundary points were firstly rotated. The rotated angle (φ_2) of each point is the derived orientation of the first fitted ellipse (φ_1), which is the angle of the ellipse major axis and the horizon axis, as shown in Fig. 4(c), so that the lower half of the ellipse is almost symmetrical about the vertical axis, as shown in Fig. 4(d). Then with the help of the bounding box location, two points corresponding to the leftmost and rightmost sides of the rotated ellipse were identified. The points within the two identified points and sitting in the lower part of the rotated ellipse were selected as the lower half boundary points. Then, they were rotated back to fit the second ellipse, as shown in Fig. 4(e), resulting in major axis a_2 and minor axis b_2 estimates. It is worth noting that when the pupil is not occluded, the two ellipses (i.e., based on all boundary points and based on the lower half boundary points) share the same orientation, major and minor axes, as shown in the left-most image in Fig. 5(b). But when a large part of the pupil is occluded, the first ellipse has opposite major and minor axes to the second ellipse, as shown in the right-most image in Fig. 5(b). In both cases, we can estimate the angle for rotation by using the first ellipse orientation.

We define a blink as the condition when the pupil is occluded by at least half since the vision begins to be suppressed, therefore, when $a_2 > 2b_1$ (different orientation of the major axis) or $|a_2 – a_1| > 0.5b_1$ (same orientation of the major axis). If the pupil is not occluded, the second ellipse is sometimes larger by a few pixels in the major axis due to the lack of top boundary point participation, however, the minor axis is similar, i.e., $|b_2 – b_1| < 3$ (pixels), as illustrated in the images in Fig. 3(a). Therefore, we define occlusion (but not blink) as $|b_2 – b_1| > 3$ (pixels) and $a_2 < 2b_1$ or $|a_2 – a_1| < 0.5b_1$.

F. Strategies to Improve the Measurement Accuracy and Processing Time

With IR light sources, the true pupil blobs always contain the highest pixel value in images, but the highest pixel value varies between video frames. In our video dataset, it ranges from approximately 0.92 to 1. After contrast adjustment, more pixels inside the pupil can be saturated to 1, which can alleviate the problem of uneven illumination on the 3-D pupil to some extent. As the binary threshold always sweeps from 1 to 0, we can reasonably assume that if there is no bounding box until the threshold of 0.95, the dark pupil does not exist in this contrast-enhanced image I_{GFC}. We can suspend the sweeping threshold and return an empty optimal threshold to speed up the processing.

However, the contrast adjustment can also cause a problem for a few images, where the pupil is fully occluded, but the eyelash area contains very high pixel values. Pixels with the highest intensity values are clustered together to form a blob and pass the blob analysis when the threshold is high (not very common). In this case, the self-tuning threshold method also returns an optimal threshold, resulting in a smoothed false blob. Another issue is the ellipse fitting itself when a substantial portion of the pupil is occluded. The lower half of the pupil boundary is too small to maintain the shape for an

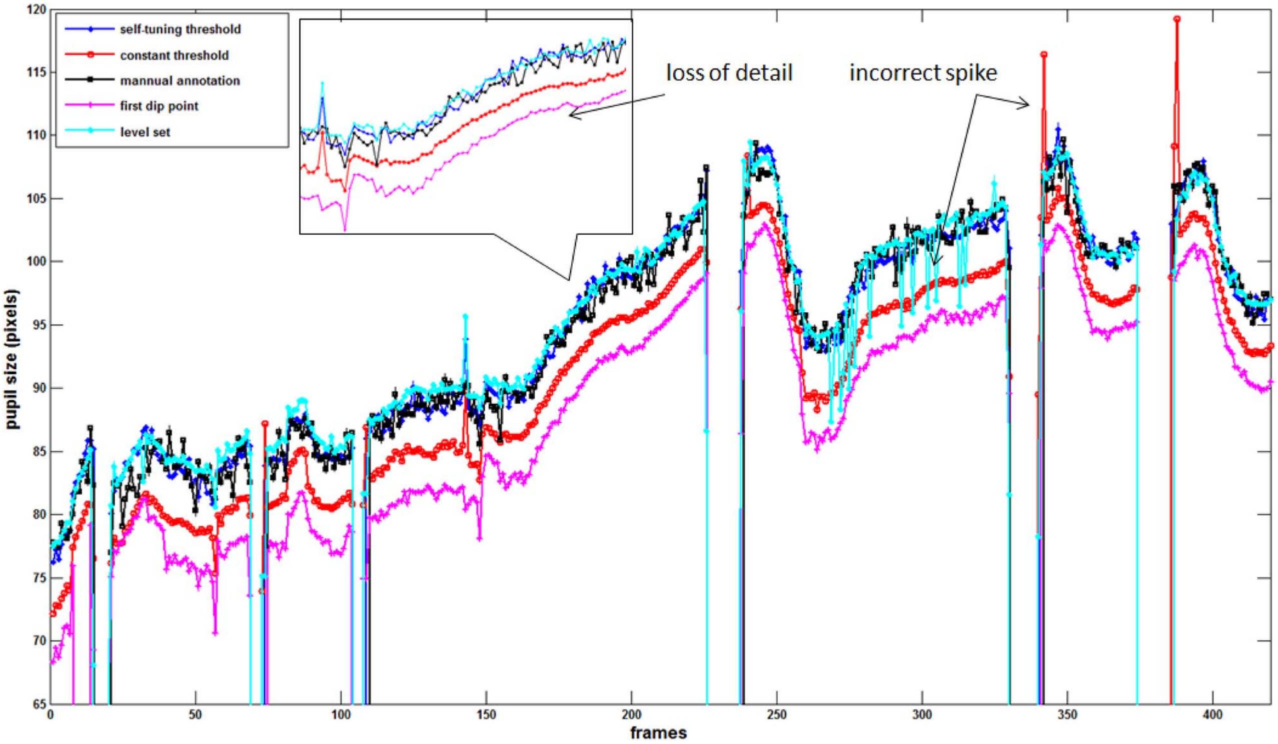


Fig. 6. Example of measured pupil size from the four methods mentioned above, compared with manual annotation during a task lasting 14 s from a video clip of the participant shown in Fig. 1(g), (i), and (k). The eye images are relatively good in the video dataset with little noise from the eyelash and eyelid. In general, the estimated size from the level set and self-tuning threshold methods were the closest to the manually annotated pupil size. However, the level set method suffers from some incorrect spikes due to low contrast regions around the pupil. The subjective constant and the first dip point threshold methods suffer from the loss of detail due to coarse thresholds.

ellipse, resulting in two similar fitted ellipses and suggesting a nonblinking frame.

Fortunately, these two issues produce a very small blob segmented by the self-thresholding method. To avoid missing blinks, an area constraint for the blob was added before ellipse fitting. With a known eye length in images, the estimated width of a small pupil is one-ninth of W_{I} (the size of the iris is about one-third of the eye length and the size of the smallest pupil is about one-third of the iris), and the height is around one-fifth of its width. Therefore, the blob area constraint was set to be around $\frac{1}{400}W_{\text{I}}$.

V. Evaluation

A. Difficult Examples

As accurate pupil size and blink occurrence measurement are of interest, we first demonstrate the effectiveness and robustness of our method in Fig. 5. Difficult examples due to the variations in pupil segmentation mentioned in Section II were selected and represent different categories of challenging pupil images. Most examples are from the eye images shown in Figs. 1 and 2. Less challenging cases, associated with high contrasts in the pupil region and less noise from the eyelid and eyelash, are also provided. They are open and fully closed eye images, as shown in the first and last example in the (a) and (c) rows in Fig. 5, respectively. Meanwhile, the measurement performance for pupil size estimation over an example task is compared with reference methods in Fig. 6. Further, failed examples from the proposed methods are also demonstrated and illustrated in Fig. 7.

B. Compared Methods

Four methods were chosen to compare with the proposed methods. They are as follows.

1) *Subjective Constant*: Subjectively select one optimal binary threshold for each video clip (similar to setting parameters in most eye tracker systems) after visual inspection and manual correction of spurious output. Morphological processing, such as opening, closing, and filling holes, was employed in order to remove the pixels resulting from eyelid shade and the notches on the pupil boundary. Dual-ellipse approaches were then used for the measurement of pupil size and blink endpoints.
2) *First Dip Point*: Manually choose a binary threshold according to the ‘first dip point’ of histogram of each frame, then use the same procedure as (1).
3) *Level Set*: Use an edge-based active contour model with the level set function in [32]. The parameter for contrast was kept constant for all compared images. The initial position was set as the bounding box expanded by 20 pixels and $\lambda = 5$ with 210 iterations; blink detection was based on the pupil existence.
4) *Proposed Self-Tuning Threshold*: The threshold t was set from 1 to 0.7 with a step of $s = 0.025$ (t can be set from 1 to 0, but no difference in E since below 0.7. This is done to reduce the number of iterations). Blinks were detected by the proposed dual-ellipse method.

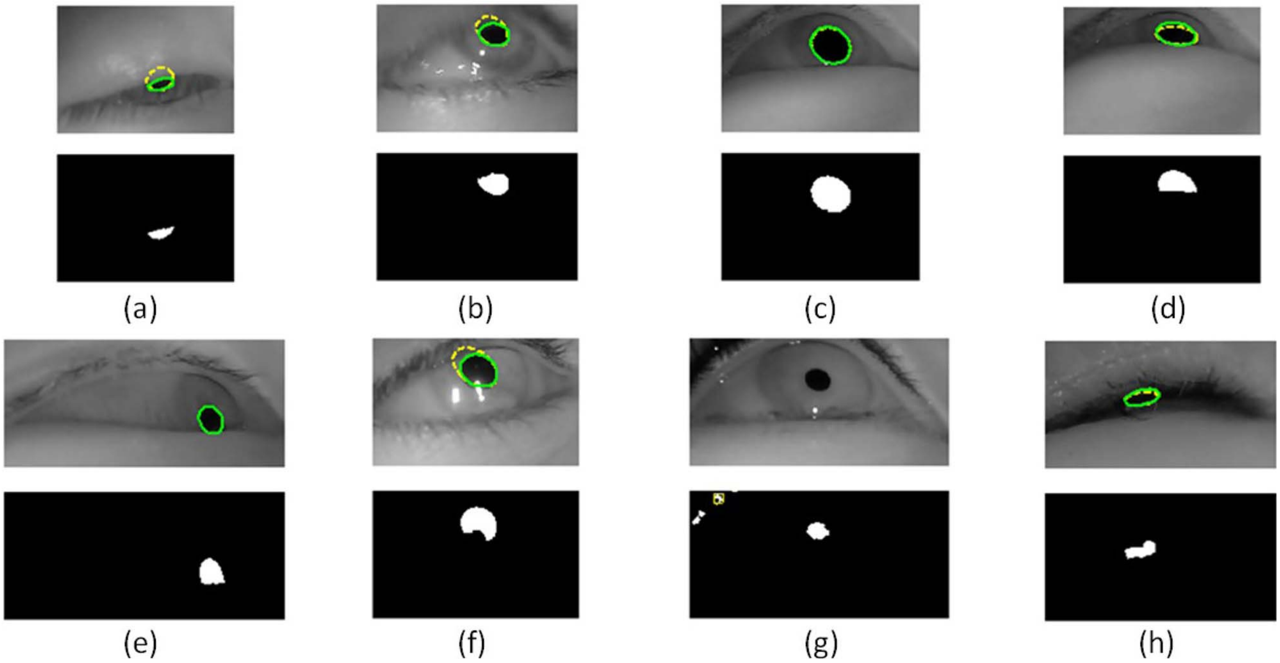


Fig. 7. Examples for which the proposed method failed (except (c) which shows the correct pupil size 1 s before (d) in order to compare the image of squint in (d)). The gray-level images with the first (green curve) and second (yellow curve) ellipses overlaid are paired with their corresponding detected pupil blobs in the binary images below them. The binary images are obtained before convex hull operation, in order to show the failure causes clearly. False blink or/and incorrect pupil size were caused by (a) gaze down for a few frames, (b) blur due to rapid eyelid motion, (d) squint causing a blocked view of the pupil, (e) failure of the second ellipse fitting due to the blocked view of the pupil, (f) glint, which affected the second ellipse fitting, (g) wrong blob detected by the first threshold, and (h) false blob due to the similar shape. Failures (a), (b) and (d) occurred around 0.001%∼0.01% of the total frames, and appeared more often than (e)-(h), which occurred around 0.0001∼0.0005 % of the total (around 4.8 million) frames.

The main reason for using the subjective constant method was due to its simplicity and speed, assuming few variations in each video clip since the webcam was fixed to the head. It is a widely used way to set parameters in eye tracking systems, and the additional processing is mainly to ensure comparable measurement accuracy. The first dip point and level set methods were included in order to compare the performance with the method proposed by De Santis and Iacoviello [14], which exactly focused on the pupil size measurement problem. Meanwhile, there is only one parameter to select in these methods (binary threshold or λ for contrast), reducing the complexity due to multiple design choices.

C. Supervised Evaluation

To investigate the performance, we randomly sampled 800 video frames (containing 17 blinking frames) from two participants' data–16 right eye videos. One participant wore heavy makeup thus had the poorest pupil size and blink estimation results, and the other is representative of the overall dataset, having an average estimation performance. The pupil size in each sampled frame was manually measured and blinking frames were manually identified. These were used as measurement references when we evaluated the four methods with multiple metrics (details are given below).

Then we used the whole video dataset to evaluate blink detection performance with the blink reference made by a visual inspection of the whole video dataset. Since it was not realistic to manually measure all pupil sizes of around 4.8 million images, evaluation of pupil size measurement across the whole dataset was not attempted. Also the first dip point and level set methods were labor intensive or computationally expensive, hence they were omitted for comparison with the whole dataset.

D. System Level Evaluation

We also performed a system level evaluation [31], in which classification accuracies of task transition, perceptual load and cognitive load were compared in order to show the correlation between the system performance and accurate pupil size measurements. Detection of task transition and nontransition segments was performed by an overlapped one-second sliding window, and three levels of perceptual load and three levels of cognitive load were classified using an overlapped two-second sliding window. The final precision and recall performance for each task were obtained by majority voting from each sliding window and evaluated with the annotated transitions or levels of each task [8]. Intuitively, the more accurate the measurement, the higher classification accuracy will be. Please note that most false blinks have been corrected for classification, so the accuracy mainly reflects the measurement of pupil size and pupil center.

E. Evaluation Metrics

Details of evaluation metrics are as follows.

1) The average mean square error (MSE) of pupil size (PS) in pixels (estimated blinking frames are removed)

$$MSE = \frac{1}{N}(PS_{ref} - PS_{measured})^2.$$

TABLE I

Comparison of Pupil Size and Blink Detection Methods, in Terms of MSE in Pupil Size, TPF, BLKFP, and BLKFN Blinks in the Subset of Videos, BDC in All Videos, TTF, PLF, and CLF

	Subjective constant (optimal)	**First dip point [14]**	**Level set [32]**	**Proposed Method**
MSE (pixel^2)	9.4	53.1	23.7	**6.5**
TPF	0.09 s (not including the time for trial and error)	10 s (manually)	12.3 s	**0.06 s**
blkFP/*blkFN* (17 blinks)	0/2	1/3	4/1	**0/0**
BDC	93.5%	-	-	**99.7%**
TTF (2-class) *PLF* (3-class) *CLF* (3-class)	65.5% 78.4% 62.1%	-	-	**67.0%** **79.7%** **64.4%**

2) Average processing time per frame (TPF): the total execution time of the proposed methods, including the self-tuning method with 13 iterations (with step 0.025), convex hull operation, boundary tracing and rotation, and dual-ellipse fitting.
3) Average number of false positive (BLKFP) and false negative (BLKFN) blink detections.
4) Blink detection accuracy (BDC) of the whole video dataset.
5) Average F_1 score (using a logistic regression classifier with a participant-dependent scheme) for classifications of task transition (TTF), perceptual load (PLF) and cognitive load (CLF). Eye features from pupil size, blink, fixation and saccade were used for classification (details see [8]).

For all methods, the processing was performed in MATLAB with a CPU @ 3.0 GHz and 4GB RAM. The results are shown in Table I.

VI. Discussion and Conclusion

As Fig. 5 shows, the self-tuning method successfully segmented the pupil from the background and correctly detected blinks in the representative easy and difficult cases without tuning parameters to cope with large variations. However, the Otsu, Iterative Threshold, Minimum Cross Entropy and Robust Automatic Threshold Selection methods could not segment the pupil from the background on these examples, as shown in Fig. 2. Fig. 5 also demonstrates that the level set method cannot work well without carefully tuning the parameter for contrast and the location for initialization. It can become trapped in local maxima when the intensity variation is not smooth enough or fail to spot the pupil boundary when the contrast between the pupil and the eyelash or eyelid is weak, as shown in Fig. 5(a) and (c). Inappropriate initial positions can also result in zero pupil size. Zero pupil size was then mistaken as a blink, which is the reason for the high false positive blink detection count in Table I.

The effectiveness of the proposed methods lies in successfully segmenting the pupil blob from the background as much as possible by utilizing the impact of higher thresholds on the location of the true pupil blob and the impact of lower thresholds on the shape of the identified pupil blob. This method did not fail to locate the correct blob in our dataset as long as one key assumption was met, the dark pupil effect produced the darkest pixels in pupil region. This lays the foundation for the subsequent processing on pupil size and blink estimation, reducing the errors due to processing on false pupil blobs, thus the robustness is improved. The effectiveness of the proposed methods can be revealed by the average MSE of the pupil size measurement in Table I. Among the four evaluation methods, the proposed self-tuning threshold achieved the closest pupil size to the reference since the MSE is significantly smaller than the other three methods. With the known physical eye lengths, the measurement accuracy is estimated to be around 0.02 mm (using the manual measurement as the ground truth) after the pixels are transformed to millimeters. Therefore, the proposed methods enable the eye tracker system to provide a full picture of the eye state for any instant in time, in terms of accurate pupil size, pupil center and blink magnitude with high autonomy.

The processing time of the proposed method is around 60 ms per frame with the aforementioned PC specification; in particular, the estimated profile execution time is 9.9 ms for the convex hull operation, 0.54 ms for the pupil blob boundary tracing and rotation, 0.66 ms for the dual-ellipse fitting, and 48.0 ms for the self-tuning step with 13 iterations (1 to 0.7 with a step of 0.025). However, if only 7 iterations were employed (1 to 0.7 with a step of 0.05), the estimated execution time for the self-tuning step was 44.0 ms. It is obvious that the processing time is significantly higher than the 33 ms frame period of the 30 Hz recording with MATLAB scripts, but we expect that it would be much faster with specific programming languages at a near-machine level to realize the real-time processing of captured images.

The proposed method has the shortest processing time in the MATLAB environment, as shown in Table I. The processing time is slightly shorter than that of the fixed subjective threshold method, which was expected to have a high speed in processing. This is because significant morphological processing, such as opening, closing, and filling holes, was required to smooth the pupil boundary since a fixed binary threshold was not perfect for each video frame. While in the proposed method, no morphological processing was used but it produced smooth pupil boundary. Furthermore, the manual intervention is incompatible with an automatic system, and the estimated cost, therefore, can hardly be bounded.

The proposed dual-ellipse method achieved a better blink detection accuracy when the pupil segmentation was more accurate. In other words, it is sensitive to the accuracy of segmentation. All blinks from the 800 frames were correctly detected. The blink accuracy for the whole dataset achieved 99.7%. It is worth noting that most failures were due to the eye looking down instead of blinking (Fig. 7(a)), and the eye reopening by nearly half between two consecutive blinks (we prefer to define this phenomenon as two blinks rather than a long blink). A small proportion of the failure was because of the blur due to fast eyelash motion, but the pupil was not occluded by half yet (shown in Fig. 7(b)), and the flat edge in the lower half pupil boundary when squinting blocked the view of the pupil in camera (Fig. 7(d) and (e)), or a large size of light reflection lay on the lower end of the pupil (Fig. 7(f)). Only a handful of blink errors occurred due to heavy makeup on the eyelid, causing higher pixel values than that in the dark pupil region (Fig. 7(g)), and the eyelash shape happened to have a pupil-like shape (Fig. 7(h)).

Using the data of pupil size and blink obtained by the proposed method, the classification performance for task characteristics was slightly better than that using the subjective optimal threshold. Among the three task characteristics, CLF significantly relies on pupil size and was improved by 2% in accuracy. This indicates fewer measurement errors in pupil size and pupil center estimation using the proposed method. These measurement errors, as shown in Fig. 6, can corrupt the estimation of task characteristics in near real-time, i.e. when eye activity is extracted over two-second segments. However, the classification accuracies achieved improved on the state of the art for classification of three load levels [8], and compare well with two-class mental state recognition [7] (to the extent that comparison is possible, since they are different problems). Although the self-tuning threshold method involves iterative computing, the number of iterations depends on the step s, indicating the resolution of threshold. This parameter can be fixed to a low value (we found 0.025 was good enough) and is not required to be tuned based on the dataset variation.

One limitation in this approach is that the identified pupil blob must contain a true pupil; otherwise, estimated results are completely wrong. Stronger IR light sources can make pupil tracking perfect, but this may be unsafe for human use. For now, we only used empirical settings to filter blobs (including an area constraint) and it may cause potential unreliability. Another limitation is that the second ellipse sometimes can cause a slightly larger ellipse than the first one when the pupil was not occluded, because of lack of the bound from the top of the pupil. Therefore it is not recommended to interpolate the pupil size during the occlusion state, instead, using the blinking state with an extension of 3 frames before and after it. Finally, the webcam is required to be placed below the eye but not too close to the face, so that the lower eyelid does not block the pupil in the view. We found that this is not difficult as all videos we collected did not have this issue. It only occurred when several participants squinted in a few video frames, which blocked the view of the pupil in the camera, resulting in a flat line on the pupil bottom, thus causing a failure in ellipse fitting.

In this paper, we investigated the problems that affect accurate pupil size and blink estimation, and proposed a novel approach to solve them. Two distinct characteristics in near-field infrared-illuminated eye images were utilized, pixels with the highest intensity always cluster together to form a pupil blob and an elliptical shape always exists in the lower half of the blob boundary. The main improvement provided by the proposed approach relative to previous work lies in dispensing with the need for parameter tuning to cope with large variations in eye videos, so that this method has the advantages of high autonomy and convenience, and is relatively efficient and robust to different environmental settings and users. Meanwhile, improved blink detection was based on current measured pupil size rather than an empirically fixed pupil size which assumes pupil size does not change. The achieved automatic load level recognition is better than the subjective and fixed threshold setting for eye activity. These suggest that the proposed methods can benefit real-time understanding of human load change for future adaptive aiding, task management, or new applications in wearable computing, whereas existing experimental configurations require manual intervention, such as calibrations for each person and thresholds for blinks, thus, lack the flexibility to operate in different environments and with different users. Future work includes improving IR filter methods or combining visible and IR images for safety and performance effectiveness issues, and further investigating the solutions for the limitations resulted from gazing down, user squinting and heavy make-up to improve robustness.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable comments to improve the paper.

References

[1] D. Li, D. Winfield, and D. J. Parkhurst, “Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches,” in *Proc. CVPR*, 2005, p. 79.

[2] Z. Zhu and Q. Ji, “Robust real-time eye detection and tracking under variable lighting conditions and various face orientations,” *Comput. Vision Image Understanding*, vol. 98, no. 1, pp. 124–154, 2005.

[3] D. W. Hansen and Q. Ji, “In the eye of the beholder: A survey of models for eyes and gaze,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 3, pp. 478–500, Mar. 2010.

[4] K. Grauman, M. Betke, J. Gips, and G. R. Bradski, “Communication via eye blinks–Detection and duration analysis in real time,” in *Proc. CVPR*, 2001, pp. I-1010–I-1017.

[5] X. Li and W. G. Wee, “An efficient method for eye tracking and eye-gaze FOV estimation,” in *Proc. ICIP*, 2009, pp. 2597–2600.

[6] X. He and P. Shi, “An efficient iris segmentation method for recognition,” in *Proc. ICAPR*, 2005, pp. 120–126.

[7] S. Chen and J. Epps, “Automatic classification of eye activity for cognitive load measurement with emotion interference,” *Comput. Methods Programs Biomed.*, vol. 110, no. 2, pp. 111–124, 2013.

[8] S. Chen, J. Epps, and F. Chen, “Automatic and continuous user task analysis via eye activity,” in *Proc. IUI*, 2013, pp. 57–66.

[9] K. F. Van Orden, W. Limbert, S. Makeig, and T. -P. Jung, “Eye activity correlates of workload during a visuospatial memory task,” *Human Factors*, vol. 43, no. 1, pp. 111–121, 2001.

[10] S. Chen and J. Epps, “Blinking: Toward wearable computing that understands your current task,” *IEEE Pervasive Comput.*, vol. 12, no. 3, pp. 56–65, Jul./Sep. 2013.

[11] J. Daugman, “Statistical richness of visual phase information: Update on recognition persons by iris patterns,” *Int. J. Comput. Vision*, vol. 45, no. 1, pp. 25–38, 2001.

[12] R. Wildes, “Iris recognition: An emerging biometric technology,” in *Proc. IEEE*, 1997, pp. 1348–1363.

[13] D. Zhu, S. T. Moore, and T. Raphan, “Robust pupil center detection using a curvature algorithm,” *Computer Methods Programs Biomed.*, vol. 59, no. 3, pp. 145–157, 1999.

[14] A. De Santis and D. Iacoviello, “Optimal segmentation of pupillometric images for estimating pupil shape parameters,” *Computer Methods Programs Biomed.*, vol. 84, no. 2, pp. 174–187, 2006.

[15] M. Sezgin and B. Sankur, “Survey over image thresholding techniques and quantitative performance valuation,” *J. Electron. Image*, vol. 13, no. 1, pp. 146–165, 2004.

[16] Q. Ji and X. Yang, "Real-time, eye, gaze, and face pose tracking for monitoring driver vigilance," *Real-Time Imaging*, vol. 8, no. 5, pp. 357–377, 2002.

[17] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst. Man Cybern.*, vol. 9, no. 1, pp. 62–66, Jan. 1979.

[18] T. W. Ridler and S. Calvard, "Picture thresholding using an iterative selection method," *IEEE Trans. Syst. Man Cybern.*, vol. 8, no. 8, pp. 630–632, Aug. 1978.

[19] A. D. Brink and N. E. Pendock, "Minimum cross-entropy threshold selection," *Pattern Recognit.*, vol. 29, no. 1, pp. 179–188, 1996.

[20] M. H. F. Wilkinson, "Optimizing edge detectors for robust automatic threshold selection: Coping with edge curvature and noise," *Graph. Models Image Process.*, vol. 60, no. 5, pp. 385–401, 1998.

[21] B. Tupper and M. Sieracki. (2013). *RATS (Robust Automatic Threshold Selection)* [Online]. Available: http://rsbweb.nih.gov/ij/plugins/rats/

[22] Z. He, T. Tan, Z. Sun, and X. Qiu, "Toward accurate and fast iris segmentation for iris biometrics," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 9, pp. 1670–1684, Sep. 2009.

[23] D. W. Hansen and A. E. C. Pece, "Eye tracking in the wild," *Computer Vision and Image Understanding*, vol. 98, no. 1, pp. 155–181, 2005.

[24] C. Pantofaru and M. Hebert, "A comparison of image segmentation algorithms," technical report, *Robotics Inst.*, Carnegie Mellon Univ., Sept. 2005

[25] R. Szeliski, *Computer Vision: Algorithms and Applications*. Berlin, Germany: Springer, 2010, pp. 270–305.

[26] P. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[27] S. Chen, J. Epps, N. Ruiz, and F. Chen, "Eye activity as a measure of human mental effort in HCI," in *Proc. IUI*, 2011, pp. 315–318.

[28] Y-L. Tian, T. Kanade, and J. F. Cohn, "Dual-state parametric eye tracking," in *Proc. Automat. Face Gesture Recognit.*, 2000, pp. 110–115.

[29] I. Bacivarov, M. Ionita, and P. Corcoran, "Statistical models of appearance for eye tracking and eye-blink detection and measurement," *IEEE Trans. Consumer Electron.*, vol. 54, no. 3, pp. 1312–1320, Aug. 2008.

[30] J. L. Crowley and F. Berard, "Multi-modal tracking of faces for video communications," in *Proc. CVPR*, 1997, pp. 640–645.

[31] H. Zhang, J. E. Fritts, and S. A. Goldman, "Image segmentation evaluation: A survey of unsupervised methods," *Comput. Vision Image Understanding*, vol. 110, no. 2, pp. 260–280, 2008.

[32] C. Li, C. Xu, C. Gui, and M. D. Fox, "Distance regularized level set evolution and its application to image segmentation," *IEEE Trans. Image Process.*, vol. 19, no. 12, pp. 3243–3254, Dec. 2010.

[33] F. Paas, J. E. Tuovinen, H. Tabbers, and P. W. M. Van Gerven, "Cognitive load measurement as a means to advance cognitive load theory," *Educ. Psychologist*, vol. 38, no.1, pp. 63–71, 2003.

[34] W. Boucsein and R. W. Backs, "Engineering psychophysiology as a discipline: Historical and theoretical aspects," *Engineering Psychophysiology: Issues and Applications*, R. W. Backs and W. Boucsein, Eds. Mahwah, NJ, USA: Lawrence Erlbaum Associates Publishers, 2000, pp. 3–30.

[35] E. Ponder and W. P. Kennedy, "On the act of blinking," *Experimental Physiology*, vol. 18, no. 2, pp. 89–110, 1927.

[36] E. H. Hess and J. M. Polt, "Pupil size in relation to mental activity during simple problem-solving," *Sci.*, vol. 143, no. 3611, pp. 1190–1192, 1964.

[37] B. P. Bailey and S. T. Iqbal, "Understanding changes in mental workload during execution of goal-directed tasks and its application for interruption management," *ACM Trans. Comput.-Human Interaction*, vol. 14, no. 4, pp. 1–28, 2008.

[38] L. Richstone, M. J. Schwartz, C. Seideman, J. Cadeddu, S. Marshall, and L. R. Kavoussi, "Eye metrics as an objective assessment of surgical skill," *Ann. Surgery*, vol. 252, no. 1, pp. 177–182, 2010.

[39] B. Laeng, S. Sirois, and G. Gredeback, "Pupillometry: A window to the preconscious?" *Perspectives Psychological Sci.*, vol. 7, no. 1, pp. 18–27, 2012.

[40] D. E. Irwin and L. E. Thomas, "Eyeblinks and cognition," in *Tutorials in Visual Cognition*, V. Coltheart, Ed. New York, NY, USA/London, U.K.: Taylor & Francis Group/Psychology Press, 2010, pp. 121–141.

[41] M. Vidal, J. Turner, A. Bulling, and H. Gellersen, "Wearable eye tracking for mental health monitoring," *Comput. Commun.*, vol. 35, no. 11, pp. 1306–1311, 2011.

[42] R. Mantiuk, M. Kowalik, A. Nowosielski, and B. Bazyluk, "Do-it-yourself eye tracker: Low-cost pupil-based eye tracker for computer graphics applications," in *Advances in Multimedia Modeling*. K. Schoeffmann, *et al.*, Eds. Berlin/Heidelberg, Germany: Springer, 2012, pp. 115–125.

[43] F. Mulvey, A. Villanueva, D. Sliney, R. Lange, S. Cotmore, and M. Donegan, "Exploration of safety issues in eyetracking," Humanities Lab., Lund University, Lund, Sweden, Rep. IST-2003-511598, 2008.

[44] E. A. Byrne and R. Parasuraman, "Psychophysiology and adaptive automation," *Biological Psychol.*, vol. 42, no. 3, pp. 249–268, 1996.

Siyuan Chen (S'11) received the M.S. degree in electronic engineering from R. M. I. T University, Melbourne, Australia, in 2008. She is currently pursuing the Ph.D. degree with the School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, Australia, and with the Machine Learning Research Group, National ICT Australia, Sydney.

Her current research interests include eye-activity computing and analysis, cognitive load modeling, machine learning for human-centered computing, and intelligent computing system.

Julien Epps (M'97) received the B.E. and Ph.D. degrees from the University of New South Wales, Sydney, Australia, in 1997 and 2001, respectively.

He was a Post-Doctoral Fellow at the University of New South Wales. He was with the Motorola Labs as a Research Engineer, where he was engaged on speech recognition and speech processing. After that, he was the Senior Researcher at the National ICT Australia, Sydney. He was a Senior Lecturer at the UNSW School of Electrical Engineering and Telecommunications, New South Wales, Australia, in 2007, where he is currently an Associate Professor. His current research interests include characterization, modelling, and classification of mental state from behavioral signals, such as speech and eye activity.

Dr. Epps has authored or co-authored around 160 publications, and regularly serves as a reviewer or technical program committee member for several IEEE journals and numerous conferences.