Link to data:
https://atreus.informatik.uni-tuebingen.de/seafile/d/8e2ab8c3fdd444e1a135/?p=%2FEyelid%20detection&mode=list

# Eyes Wide Open? Eyelid Location and Eye Aperture Estimation for Pervasive Eye Tracking in Real-World Scenarios


**Wolfgang Fuhl**
Perception Engineering
University of Tübingen, Germany
wolfgang.fuhl@uni-tuebingen.de

**Thomas Kübler**
Perception Engineering
University of Tübingen, Germany
thomas.kuebler@uni-tuebingen.de

**Thiago Santini**
Perception Engineering
University of Tübingen, Germany
thiago.santini@uni-tuebingen.de

**Wolfgang Rosenstiel**
Computer Engineering
University of Tübingen, Germany
wolfgang.rosenstiel@uni-tuebingen.de

**David Geisler**
Perception Engineering
University of Tübingen, Germany
david.geisler@uni-tuebingen.de

**Enkelejda Kasneci**
Perception Engineering
University of Tübingen, Germany
enkelejda.kasneci@uni-tuebingen.de




Every submission will be assigned their own unique DOI string to be included here.



## Abstract
Eyelid identification and aperture estimation provide key data that can be used to infer valuable information about a subject's mental state (e.g., vigilance, fatigue, and drowsiness) as well as validate or reduce the search space of other eye features. In this paper, we consider these tasks from the perspective of pervasive eye tracking, taking into account the multiple challenges and constraints that arise from this scenario. A novel method for eyelid identification and aperture estimation is proposed and evaluated against challenging data from an eye-tracking experiment conducted in driving scenarios in the wild. The proposed method outperforms an state-of-the-art approach by up to 40 percentage points and runs in real-time on state-of-the-art eye tracking systems. The method implementation and the realistic dataset are provided openly at www.ti.uni-tuebingen.de/perception .



## Author Keywords
Eyelid detection; Pervasive eye tracking; Eyelid closure; Image processing; Wearable computing; Real time; Dataset

## ACM Classification Keywords
I.5.4 [PATTERN RECOGNITION]: Applications; I.4.8 [IMAGE PROCESSING AND COMPUTER VISION]: Object recognition,Shape; I.4.9 [IMAGE PROCESSING AND COMPUTER VISION]: Applications

**Figure 1:** Multiple levels of pupil occlusion during a blink.

## Introduction

The human eyelid is a skin fold that performs multiple tasks with the intent of protecting the anterior surface of the ocular globe. Among these tasks, the eyelids aid in regulating the amount of light that reaches the eye (e.g., by squinting when gazing at bright light sources) as well as eye maintenance (e.g., by distributing the tear film over the cornea during blinks) [6, 8]. Although essential for perpetuating eye healthiness, these tasks alter the eye appearance significantly, posing a myriad of problems for the identification of relevant eye features in video-based eye trackers. For instance, they may lead to several levels of pupil and iris occlusion (e.g., see Figure 1) or obstruct their illumination. Furthermore, the eyelids may continually occlude these eye features depending on the eye position with respect to the camera. Therefore, an accurate and robust method for estimating the position of the eyelids can be employed, for instance, to validate (or even reduce the search space for) features of interest. Moreover, the correct estimation of eyelid aperture also has direct applicability for blink, vigilance, fatigue, and drowsiness identification [17, 27, 24].

In this work, we focus on head-mounted video-based eye trackers since these are relatively unintrusive, flexible, and mobile, thus making them excellent candidates for pervasive eye tracking. These eye trackers provide eye images directly, thus no face detection nor eyes boxes localization are required. However, such trackers impose some additional constraints on the task. For example, color images are not available due to the use of the near-infrared imaging, and the eye orientation as well as eye canthi position are not known a priori. Furthermore, a direct consequence of pervasive eye tracking is the usage of the system in real-world and natural scenarios. In contrast to laboratory conditions, these scenarios further increase the complexity of detecting eye features due to the introduction of several sources of noise – e.g., complex/changing illumination, motion blur, recording errors, occlusion by eyelashes, and reflections [10]. Furthermore, some of the practical applications for these devices must be in the form of embedded systems (e.g., driving assistance [2, 12]) and impose real-time, processing, and energy consumption constraints.

To the best of our knowledge, this is the first work to approach eyelid detection and aperture estimation from the aspect of pervasive eye tracking; additionally, the proposed method runs in real-time and does not require the detection of other eye features. Our main contributions are:

- A novel method to identify pixels that approximate the lower and upper eyelids based on a likelihood map derived from four features computed from a grayscale close-up image of the eye region.
- Alternatively to fitting polynomials to the pixels that approximate the eyelid, we propose approximating the eyelids outline using the ending points of edges extracted from this likelihood map to fit two Bézier curves, forming an ellipsis. The minor axis of this ellipsis is used as an estimate of the eye aperture.
- We introduce a new challenging dataset containing 1100 eye images collected during eye-tracking experiments from on-road driving. Each image was annotated by experts with ten roughly equally-spaced eyelid points per eye image, including eye canthi. To foster further research, this dataset is provided openly at www.ti.uni-tuebingen.de/perception .
- The proposed solution is evaluated by comparing the estimated eyelid aperture to the aperture derived from the points annotated by the experts, improving the eyelid aperture estimation up to 40 percentage points relative to a state-of-the-art approach.

**The Competition**

In this work, we have opted to evaluate the proposed algorithm against *VASIR* [14].

**Advantages:** Besides its status as an state-of-the-art project, it is also an open-source one, thus avoiding introducing misleading results due to small reimplementation differences; it is often the case that not all details are included in papers due to space limitations.

**Similarities:** Moreover, several approaches, such as [1, 3, 11, 15, 16, 18, 19], follow similar procedures of first finding the iris/pupil and then approximating the contours of the eyelids based on the top and bottom regions around the iris. Thus, *VASIR*'s, through its multiple linear Hough transforms to locate the eyelids, is a robust representative of this class of approaches in terms of both results and run time.

## Related Work

The task of eyelid aperture estimation is closely related to the detection of the eyelids. Approaches tackling eyelid detection stem as a byproduct from two particular tasks, namely, blink detection and iris recognition; the former being usually applied in vigilance, fatigue, and drowsiness detection [17, 24, 27]. McIntire et al. [17] report that the *Eye-Com* alertness monitoring device identifies blinks when more than 85% of the pupil is occluded by the eyelid; however, no information is given on how the pupil and eyelid detections are achieved. Yang et al. [27] apply a deformable eye contour template based on two parabolic sections intersecting at two points (the eye corners); this template is then deformed to maximize a likelihood function based on the clustering of color and texture descriptors. Suzuki et al. [24] divide the input image into vertical sections and, for each section, find the pair of maximal and minimal intensity derivatives most distant from the darkest point in the section. The candidates from five sections are grouped, and two groups are estimated to represent the upper and lower eyelid; the average distance between the upper and lower eyelid points is calculated and used as the *eye gap* metric. In the context of iris recognition, eyelid detection is employed to improve iris segmentation, leading to improved recognition rates [4]. For instance, the Wildes method [26] employs Hough transforms for parabolic arcs to identify the eyelids, and Daugmann [4] uses a integrodifferential operator with arcuate contour integration, in which spline parameters are fitted based on statistical estimation. Additionally, some methods rely on the a priori identification of other eye features (e.g., pupil, iris, canthi), which is not a trivial task for images obtained in real-world and unconstrained scenarios. Cai and Wang [3] first preprocess the input image with morphological operations to remove eyelashes and light spots. The minimum grayscale value of each column is then extracted as edges, and a least square parabolic fitting is applied to these edges. The result is then refined to compensate for deviations due to the preprocessing. Radman et al. [23] first utilize a radial edge detector based on the outer iris boundary; for each pixel, the saturation average value of its ten adjacent pixels is calculated and, once this average exceeds a threshold, the pixel is considered an intersection point between the eyelid and outer iris boundary. This results in starting and ending pixels for each eyelid, which are then connected through the live wire method [19]. Adam et al. [1] initially smooth the input image through anisotropic diffusion and apply a Canny edge detector. The search area in the edge image is then restricted to the top and bottom of the inner iris. The edges in the region of interested are then filtered based on the mean length of all edges in the region; parabolas are fitted to remaining edges, and their direction used for further filtering. The remaining edge that maximizes the horizontal Sobel gradient intensity from the original image is chosen as contour. The *VASIR* (Video-based Automatic System for Iris Recognition) is a state-of-the-art iris recognition platform developed by the National Institute of Standards and Technology [14]. In VASIR, eyelid detection starts by locating the pupil and iris through Hough transforms. Afterwards, the region around the upper part of the iris is split into three parts. The eyelid in each subregion is found using a linear Hough transform, and a third-degree polynomial is fit to these points through Lagrange interpolation; the same procedure is repeated for the lower eyelid.

## Eyelid Detection

Let $I[r, c]$ be a digital close-up image of the eye in the near-infrared spectrum with $r$ rows and $c$ columns. The eyelid detection task consists of selecting two sets of pixels $P_l$ and $P_u$ in $I$ that lie respectively on the *lower* and *upper* eyelids, which are then used to fit functions representing the outline of each eyelid.

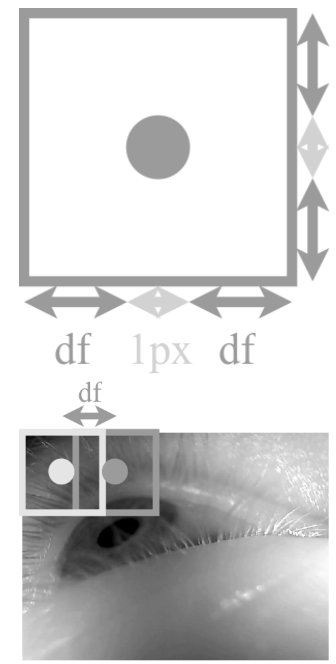



**Figure 2:** Downscaling window size (on the top) and stride (on the bottom) – not in scale in relation to each other.

The proposed method consists mainly of I) rescaling the image preserving dark regions to reduce noise and computation costs, II) filtering the image according to a combination of local features to generate a likelihood map for the eyelids, III) detecting edges on the likelihood map, and selecting two edges to represent the eyelids based on their orientation and horizontal shift in respect to one another, enclosed intensity value, and accumulated likelihood. These steps are described in detail in the following subsections, followed by a graphical representation exemplifying the output of each stage in the algorithm (Figure 4).

*I - Rescaling*

To preserve thin dark structures that usually lie close to the eyelid, such as eyelashes, we employ a downscaling operation that favors lower intensity pixels (as proposed in [10]). Let $df$ be a downscaling factor (five in our implementation). The values of the downscaled image $D$ are calculated from the pixels $p_i \in I$ based on a square sliding window $W$ with sides of $l = 2 * df + 1$ pixels and stride of $df$ pixels (see Fig. 2). For each position of $W$, the mean intensity in the window is calculated as

$$\mu = \frac{1}{l^2} \sum_{p_i \in W} p_i, \tag{1}$$

the window intensity histogram $H$ is computed, and the corresponding pixel value $d$ in the downscaled image is then evaluated as

$$d = \left( \sum_{j=0}^{\mu} H(j).j \right) \Bigg/ \left( \sum_{j=0}^{\mu} H(j) \right). \tag{2}$$

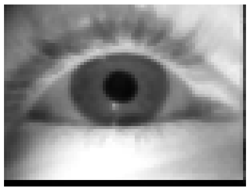
*II - Likelihood Map Generation*

Four different local features are exploited to generate the likelihood map, namely the mean, standard deviation, skewness, and horizontal edges. For each pixel in $D$, these features are derived from a neighborhood centered on that pixel. The mean, standard deviation, and skewness are computed over a local square window sized $7 \times 7$ pixels and act as rotation invariant sparse edge filters; additionally, these filters also respond to edges partially covered by eyelashes. The mean is calculated as the first moment ($m_1$) and uses the complement of the downscaled image as input (to weight small shadows close to the eyelids higher). The standard deviation is calculated as the second moment around the mean ($m_2$), whereas the skewness is evaluated as the third moment around the mean ($m_3$). The horizontal edge response is calculated using the Prewitt operator [21] and serves as a reinforcement of horizontal edges in the likelihood map; it is worth noticing that even if the eye is not precisely aligned horizontally with relation to the camera, some parts of the eyelid evoke a response due to the arcuate nature of the eyelid. Each feature produces an associated activation map: the mean ($M_1$), standard deviation ($M_2$), skewness ($M_3$), and horizontal edge ($E$) maps. These maps are point-wisely[1] combined to generate the likelihood map $L$ as

$$L = E^2 \odot M_1 \odot M_2 \oslash M_3, \tag{3}$$

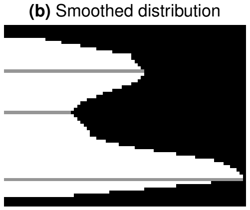
effectively resulting in a high pass filter for $E$, $M_1$ and $M_2$ and in a low pass one for $M_3$. An averaging filter with a height of three pixels and covering $\approx 30\%$ of $L$'s width is applied to $L$ in order to connect horizontally disjointed high likelihood regions and increase the response of straight horizontal edge parts. This operation introduces some noise, which is partially removed by setting negligible values (smaller than one cent of one percent of the maximum value) to zero. Additionally, we set likelihood map values outside plausible eyelid regions to zero. The boundaries of

[1] $\odot$ and $\oslash$ denote point-wise multiplication and division, respectively.

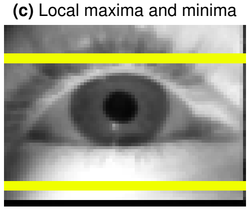


**(a)** Rescaled $(D)$

**(b)** Smoothed distribution



**(c)** Local maxima and minima



**(d)** Boundaries

**Figure 3:** The input (a), and its smoothed mean horizontal intensity values distribution (b), from which local maxima and minima (c) are identified and employed to determine the plausible eyelid boundaries – yellow lines in (d).

this region are determined based on the mean horizontal intensity values distribution, considering that the pupil and iris exhibit lower intensity values relative to the skin patches above and below it. Prior to the analysis, the distribution is smoothed with an averaging filter of seven pixels to remove high peeks and holes (Figure 3b). Afterwards, all local maxima and minima are determined (Figure 3c); the tuple of two distinct maxima and one minimum $\{max_a, max_b, min\}$ that maximizes the distance

$$max_a + max_b - 2 \cdot min \tag{4}$$

is used to determined the boundaries, which are set to $max_a$ and $max_b$ (Figure 3d). This yields the refined likelihood map $L_r$.

*III - Edge Detection and Selection*
Edge detection is applied to $L_r$ by means of non-maximum suppression, followed by a thinning morphological operation, resulting in a set of edges. Let $E_i$ and $E_j$ be two distinct edges, and the mean position for an edge be the mean position of all pixels belonging to the edge. For each possible pair of edges $(E_i, E_j)$, four metrics are calculated:

$\Sigma_L(E_i, E_j)$ **:** The *accumulated likelihood* is based on the values $v_k \in L_r$ and defined as

$$\Sigma_L(E_i, E_j) = \sum_{v_k \in E_i} \times \sum_{v_k \in E_j} \tag{5}$$

$\delta_h(E_i, E_j)$ **:** The *horizontal shift* is defined as the distance between the horizontal components of the edges mean position.

$\alpha(E_i, E_j)$ **:** The *relative angle* is defined as the normalized angle between the mean position of the edges (e.g., $90° \overset{\alpha}{\mapsto} 1$ and $0° \overset{\alpha}{\mapsto} 0$).

$\iota(E_i, E_j)$ **:** The *enclosed intensity* is defined as the mean intensity enclosed by the area generated by the seven pixels orthogonal to each pixel in the vector that connects the mean position of the edges.

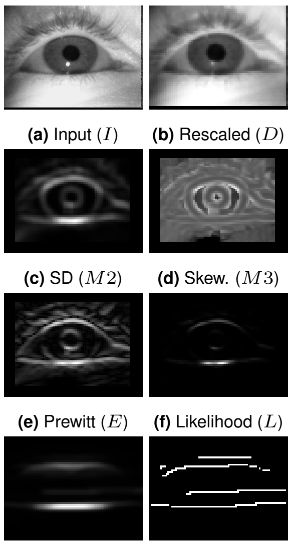
Figure 6 shows a graphical representation of these metrics for an edge pair. For each pair, these metrics are then combined to form a total score

$$\tau(E_i, E_j) = \Sigma_L \times \delta_h \times \alpha \times \iota, \tag{6}$$

and the pair with highest score is selected as eyelids. Afterwards, the selected edge points with values in $L_r$ smaller than one third of their maximum value are removed to attenuate the effect of spurious edges introduced by the filters in previous steps.

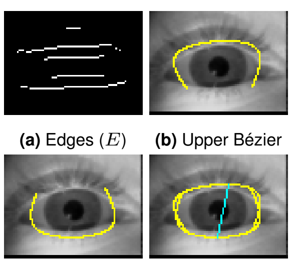
## Eyelid Aperture Estimation

In order to estimate the eyelid aperture, the ending points of the upper and lower eyelids are used to fit two Bézier curves. One curve uses the upper eyelid ending points as first and last control points, whereas the other use those from the lower eyelid. The combination of these two curves result in an ellipsis that approximates the eye outline. The major and minor ellipsis axis are determined based on the two orthogonal point pairs with maximal distance, and the minor axis is used as estimate for the eyelid aperture. To compensate for the small vertical smearing introduced by the box filter in the previous step, we substract two pixels (one for each eyelid) from the estimated distance before upscaling it to the input image scale. Figure 5 shows the resulting procedure for three distinct situations. The main advantage of the proposed Bézier-curve-based approach is that fitting the Bézier curves is a stable procedure, whereas commonly polynomial fit approaches employed in related work are unstable. As a result, the algorithm performs more uniformly across different scenarios. It is important to notice



**(a)** Input ($I$) **(b)** Rescaled ($D$)
**(c)** SD ($M2$) **(d)** Skew. ($M3$)
**(e)** Prewitt ($E$) **(f)** Likelihood ($L$)
**(g)** Refined ($L_r$) **(h)** Edges ($E$)

**Figure 4:** Function performed by each stage in the eyelid detection algorithm – normalized per image.



**(a)** Edges ($E$) **(b)** Upper Bézier
**(c)** Lower Bézier **(d)** Aperture

**Figure 5:** Edge ($E$), upper and lower eyelid Bézier curves, and the resulting ellipsis with the aperture estimation (minor axis, in cyan).

that this approach does not model the eye canthi region accurately; however, these regions can be safely disregarded without loss of information since they are not pertinent to estimating the eyelid aperture and features of interest (e.g., pupil) do not lay in these regions.

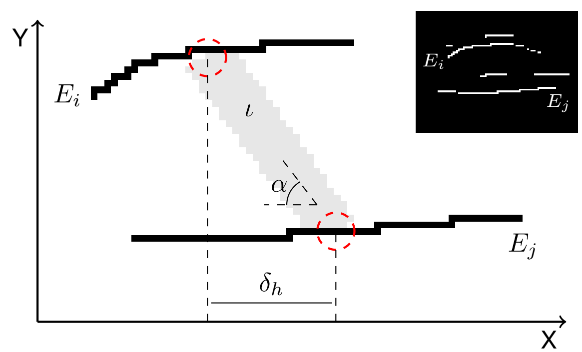



**Figure 6:** Graphical representation of edge selection metrics for an edge pair $(E_i, E_j)$. Red dashed circles present the mean position of the edges, and the gray area represents the area considered when evaluating $\iota$. This is the edge pair selected to represent the eyelids given the edge image on the top right corner.

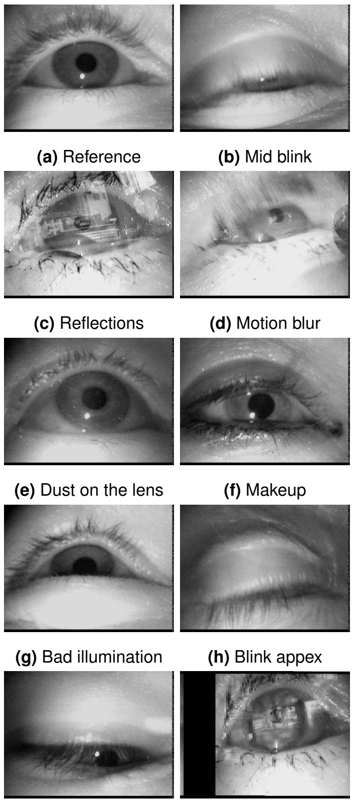
## Evaluation

In order to compare the proposed approach to the state-of-the-art, we have employed the method from the VASIR [14] project to detect eyelids, combined with the "*eye gap*" measure [24]. During this evaluation, we have found the third-degree polynomial employed by VASIR to approximate the eyelids to be relatively unstable in some cases; as such, we additionally evaluate the same method but replacing the third-degree polynomial by a second-degree one. Henceforth, the former is referred to as *VASIR-3d* and the latter as *VASIR-2d*.

*Dataset*

As previously mentioned, eyelid detection is mostly related to two research areas. For vigilance, fatigue, and drowsiness detection, data is recorded to assess those particular metrics, and no eyelid ground-truth is present. To the best of our knowledge, none of these datasets are publicly available. For the second field, namely iris recognition, there are multiple publicly available datasets containing eyelid data annotation – although they often require lengthy registration processes a priori. The most widespread datasets in the literature are UBIRIS [22] and CASIA [20] datasets. However, these datasets are designed for biometry purposes and, thus, often follow guidelines to ensure that the amount of information existent in the data and the proportion of noise occluding the iris meet certain requirements – e.g., iris diameter of at least 140 pixels [4]. As a result, these data are not realistic when pervasive eye tracking is considered.

To circumvent these issues, we have collected and manually labeled $1100$ images (384x288 pixels) from an eye-tracking experiment conducted during driving in real-world scenarios [13]. These data stem from eleven subjects and include several challenging situations that present themselves in pervasive eye tracking scenarios such as mid-blink images, reflections, motion blur, dust on the lens, makeup usage, bad illumination due to camera angle, fully closed eyes, eye lashes, and even rare artifacts (see Figure 7). Per subject, a hundred frames were randomly sampled respecting a minimum temporal distance between frames as to increase frame dissimilarity while keeping labeling efforts tractable. Each frame was annotated with ten roughly equally-spaced points: one point on each eye can-



**(a)** Reference **(b)** Mid blink **(c)** Reflections **(d)** Motion blur **(e)** Dust on the lens **(f)** Makeup **(g)** Bad illumination **(h)** Blink appex **(i)** Eye lashes **(j)** Artifact

**Figure 7:** A good quality frame for reference and challenging situations pertinent to pervasive eye tracking.

thus, four points lying on the lower eyelid, and four points lying on the upper eyelid. To foster further research on the topic, we provide these datasets openly and straightforwardly at www.ti.uni-tuebingen.de/perception .

*Eyelid Aperture Estimation*

As a measure of eyelid aperture, we utilize the palpebral fissure height (PFH) [7]. To derive the PFH from the ground-truth, first each eyelid outline is determined by fitting Bézier splines to the annotated eyelid points (including canthi). Next, the palpebral fissure width (PFW) is estimated by tracing a vector connecting the eye canthi. Then, the PFH is determined by finding the largest vector orthogonal to the PFW vector and delimited by the eyelids (see Figure 10). The estimation error $\varepsilon$ is then evaluated as

$$\varepsilon = PFH - \omega_{alg}, \tag{7}$$

where $\omega_{alg}$ is the aperture estimate from the algorithm. It is important noticing that humans are known not to produce pixel-accurate ground-truth, and often an error range of five pixels is considered for a single feature position (e.g., the pupil center) in the literature [10, 9, 25]. Thus, we discuss our results up to an error of ten pixels due to the intrinsic error introduced by the annotators.

Figure 8 reports the cumulative detection rate for the eyelids aperture per estimation error, clearly showing that the proposed method outperforms both *VASIR2d* and *VASIR3d* – improving the correct estimation of the eyelid aperture up to 40 percentage points relative to the state-of-the-art approaches for the ten pixels tolerance. This figure also reveals that the second and third degree polymons produced similar estimations for the eyelid aperture.

To further investigate the performance of the algorithms, we break down the estimation error per subject in Figure 11.

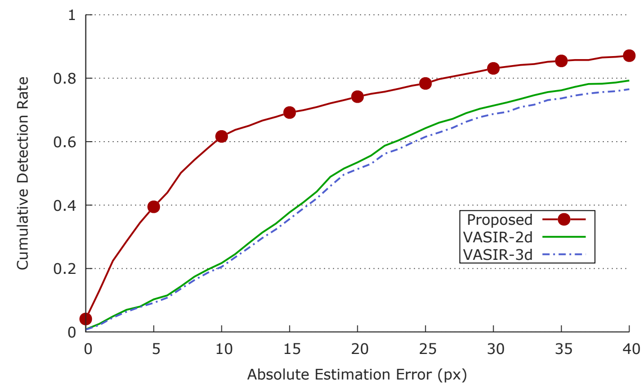



**Figure 8:** Eyelid aperture cumulative detection rate.

Results are reported using *boxplots* – a box is drawn between the first and third quartiles, a horizontal line represents the median value, and whiskers extend to the minimum and maximum values. This figure shows that the proposed algorithm performed better and with less variance than the state-of-the-art approaches in the majority of cases – the exceptions being subjects $6$ and $7$, where *VASIR-2d* performed slightly better, and subject $11$, where none of the approaches performed satisfactorily. As can be seen in Figure 9, subject $6$ wears very dark make-up around the eye, which the proposed algorithm wrongly identified as the eyelid. Similar behavior happened for subject $7$ due to slight ptosis combined with large creases in the skin of the upper eyelid. The subpar performance for Subject $11$ is easily explained by the quantity and size of bright reflections covering most of the eye image; this is by far the most challenging subject in the contributed dataset.

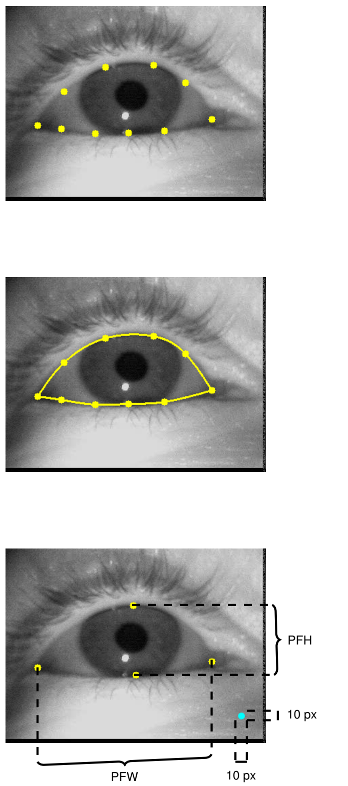



**Figure 10:** Deriving the PFH reference value from the ground truth. The annotated ground-truth eyelid points (top) are fitted with Bézier splines (middle), and the longest vector orthogonal to the PFW delimited by the eyelids is selected as PFH (bottom). As reference for the error tolerance, a circle with diameter of ten pixels is also shown (in cyan).

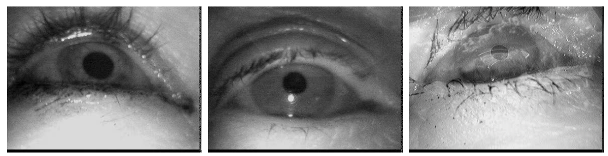


**Figure 9:** Samples respectively from subjects 6, 7, and 11.

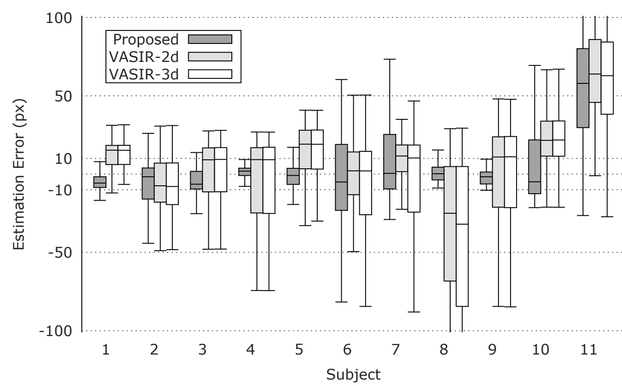



**Figure 11:** Eyelid aperture estimation error for each subject.

It's worth noticing that simply producing a good estimation for the eyelid aperture does not suffice. An algorithm may be simply detecting something else that happens to coincide with the eyelid aperture. To account for this, we also assess the algorithms in terms of segmenting the inter-eyelids region. The ground-truth segmentation is given by the inner area of the Bézier-spline-connected points from annotations. As similarity metric, we employ the popular Dice similarity coefficient [5]; the closer the value is to one, the better. The results are presented in Figure 12 and show that the algorithms performed fairly well in the segmentation task, even though neither models the eye canthi appropriately. *VASIR-2d* produced more accurate segmentations, which one can expect since it operates on the full scale image, whereas the proposed algorithm operates on a downscaled image. Nonetheless, the proposed approach produced more consistent results. This figure also shows that the second-degree polynomial fit better to the data than the third-degree polynomial. The *VASIR* outliers close to zero stem from blink images, in which the pupil/iris detection step failed to locate an area inside the eyelid region.

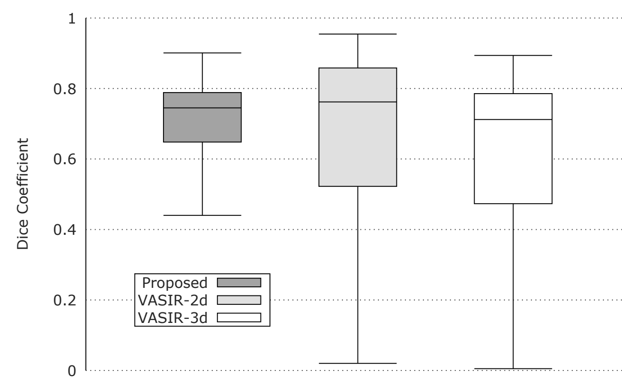



**Figure 12:** Dice coefficient for inter-eyelid region segmentation.

Algorithm run time was evaluated on a Intel® Core™ i5-4590 CPU @ 3.30GHz with 3GB RAM, under Ubuntu 14.04.02 running inside a virtual machine, which is similar to systems employed by vendors. Results are reported in Table 1 and demonstrate the superiority of the proposed approach in terms of required execution time. *VASIR*'s run time is roughly broken down as pupil detection $(37.26\%)$, iris de-

| | $\mu(ms)$ | $\sigma(ms)$ |
|---|---|---|
| Proposed | 3.15 | 0.32 |
| VASIR-2d | 404.84 | 158.56 |
| VASIR-3d | 404.81 | 158.57 |

**Table 1:** Mean ($\mu$) and standard deviation ($\sigma$) for the execution time for each of the evaluated algorithms based on all images in the dataset.

**Employabillity in State-of-the-Art Eye Trackers**

Due to the high run time of *VASIR*, that method is not employable in real-time even for the slowest head-mounted eye trackers, which tend to have a resolution of 25 frames per second, requiring the algorithm to perform under 40 ms. On the contrary, the proposed method is employable even for higher-end head-mounted eye trackers running at 120 frames per second (i.e., 8 ms) such as the SMI Eye Tracking Glasses 2.

tection $(47.8\%)$, linear hough transform $(14.86\%)$, and Lagrange interpolation $(0.08\%)$.

## Conclusion

In this paper, we have proposed a fast and robust method to estimate the region around the eyelids and eyelid aperture. The proposed approach reaches more than 60% percent detection rate on a challenging and realistic dataset, outperforming a state-of-the-art method by more than 40%. Whereas the algorithm performed well for most challenges that can be expected in pervasive eye tracking scenarios, performance for large and strong reflections was not satisfactory – although smaller reflections and light halos did not interfere with the performance. Nonetheless, further work is required to deal with these reflections; particularly because these are rather common in the presence of glasses, which are worn by a considerable segment of the population. The proposed algorithm implementation and the datasets are available at www.ti.uni-tuebingen.de/perception .

## REFERENCES


1. M. Adam, F. Rossant, F. Amiel, B. Mikovikova, and T. Ea. 2008. Eyelid localization for iris identification. *Radioengineering* 17, 4 (2008), 82–85.
2. C. Braunagel, E. Kasneci, W. Stolzmann, and W. Rosenstiel. 2015. Driver-Activity Recognition in the Context of Conditionally Autonomous Driving. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. 1652–1657.
3. P. Cai and C. Wang. 2015. An eyelid detection algorithm for the iris recognition. *International Journal of Security and Its Applications* 9, 5 (2015), 105–112.
4. J. Daugman. 2004. How iris recognition works. *Circuits and Systems for Video Technology, IEEE Transactions on* 14, 1 (2004), 21–30.
5. L. R Dice. 1945. Measures of the amount of ecologic association between species. *Ecology* 26, 3 (1945), 297–302.
6. PN Dilly. 1994. Structure and function of the tear film. In *Lacrimal Gland, Tear Film, and Dry Eye Syndromes*. Springer, 239–247.
7. S. A Fox. 1966. The palpebral fissure. *American journal of ophthalmology* 62, 1 (1966), 73–78.
8. FB Fralick. 1962. Anatomy and physiology of the eyelid. *Transactions-American Academy of Ophthalmology and Otolaryngology. American Academy of Ophthalmology and Otolaryngology* 66 (1962), 575.
9. W. Fuhl, T. Kübler, K. Sippel, W. Rosenstiel, and E. Kasneci. 2015. ExCuSe: Robust Pupil Detection in Real-World Scenarios. In *Computer Analysis of Images and Patterns*. Springer, 39–51.
10. W. Fuhl, T. C. Santini, T. Kübler, and E. Kasneci. 2016. ElSe: Ellipse Selection for Robust Pupil Detection in Real-world Environments. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications (ETRA '16)*. ACM, New York, NY, USA, 123–130.
11. B. J. Kang and K. R. Park. 2007. A robust eyelash detection based on iris focus assessment. *Pattern Recognition Letters* 28, 13 (2007), 1630–1639.
12. E. Kasneci, G. Kasneci, T. C. Kübler, and W. Rosenstiel. 2015. *Artificial Neural Networks: Methods and Applications in Bio-/Neuroinformatics*. Springer, Chapter Online Recognition of Fixations, Saccades, and Smooth Pursuits for Automated Analysis of Traffic Hazard Perception, 411–434.

13. E. Kasneci, K. Sippel, K. Aehling, M. Heister, W. Rosenstiel, U. Schiefer, and E. Papageorgiou. 2014. Driving with Binocular Visual Field Loss? A Study on a Supervised On-road Parcours with Simultaneous Eye and Head Tracking. *Plos One* 9, 2 (2014), e87470.

14. Y. Lee, R. J Micheals, J. J Filliben, and P J. Phillips. 2013. VASIR: an open-source research platform for advanced iris recognition technologies. *Journal of research of the National Institute of Standards and Technology* 118 (2013), 218.

15. X. Liu, K. W. Bowyer, and P. J. Flynn. 2005. Experiments with an improved iris segmentation algorithm. In *Fourth Workshop on Automatic Identification Advanced Technologies*. IEEE, 118–123.

16. T. Mäenpää. 2005. An iterative algorithm for fast iris detection. In *Advances in Biometric Person Authentication*. Springer, 127–134.

17. L. K. McIntire, R. A. McKinley, C. Goodyear, and J. P. McIntire. 2014. Detection of vigilance performance using eye blinks. *Applied ergonomics* 45, 2 (2014), 354–362.

18. T. Min and R. Park. 2009. Eyelid and eyelash detection method in the normalized iris image using the parabolic Hough model and Otsu's thresholding method. *Pattern recognition letters* 30, 12 (2009), 1138–1143.

19. E. Mortensen, B. Morse, W. Barrett, and J. Udupa. 1992. *Adaptive boundary detection using 'live-wire' two-dimensional dynamic programming*. IEEE.

20. Chinese Academy of Sciences' Institute of Automation. 2016. CASIA Iris Image Database. `http://biometrics.idealtest.org`. (2016).

21. J. MS. Prewitt. 1970. Object enhancement and extraction. *Picture processing and Psychopictorics* 10, 1 (1970), 15–19.

22. H. Proença, S. Filipe, R. Santos, J. Oliveira, and L. A. Alexandre. 2010. The ubiris. v2: A database of visible wavelength iris images captured on-the-move and at-a-distance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 32, 8 (2010), 1529–1535.

23. A. Radman, N. Zainal, and M. Ismail. 2013. Efficient iris segmentation based on eyelid detection. *Journal of Engineering Science and Technology* 8, 4 (2013), 399–405.

24. M. Suzuki, N. Yamamoto, O. Yamamoto, T. Nakano, and S. Yamamoto. 2006. Measurement of driver's consciousness by image processing-a method for presuming driver's drowsiness by eye-blinks coping with individual differences. In *Systems, Man and Cybernetics, 2006. SMC'06. IEEE International Conference on*, Vol. 4. IEEE, 2891–2896.

25. L. Świrski, A. Bulling, and N. Dodgson. 2012. Robust real-time pupil tracking in highly off-axis images. In *Proc. of the Symposium on ETRA*. ACM, 173–176.

26. R. P. Wildes. 1997. Iris recognition: an emerging biometric technology. *Proc. IEEE* 85, 9 (1997), 1348–1363.

27. F. Yang, X. Yu, J. Huang, P. Yang, and D. Metaxas. 2012. Robust eyelid tracking for fatigue detection. In *2012 19th IEEE International Conference on Image Processing*. 1829–1832.