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

## Fast and Robust Eyelid Outline and Aperture Detection in Real-World Scenarios

Wolfgang Fuhl University of Tübingen Thiago Santini University of Tübingen thiago.santini@uni-tuebingen.de

wolfgang.fuhl@uni-tuebingen.de

Enkelejda Kasneci University of Tübingen

Enkelejda.Kasneci@uni-tuebingen.de

#### Abstract

The correct identification of the eyelids and its aperture provide essential data to infer a subject's mental state (e.g., vigilance, fatigue, and drowsiness) and to validate or reduce the search space of other eye features (e.g., pupil, and iris). This knowledge can be used not only to improve many applications, such as eye tracking and iris recognition, but also to derive information about the user (such as, the take-over readiness of the driver in the automated driving context). In this paper, we propose a computervision-based approach to eyelids identification and aperture estimation. Evaluation was performed on an existing data set from the literature as well as on a new data set introduced in this work. The new data set contains 4000 hand-labeled eye images from 11 subjects driving in a city; these contain several challenges such as reflections, makeup, wrinkles, blinks, and changing illumination. The proposed method outperformed state-of-the-art methods by up to  $\approx 16.11$  percentage points in terms of average similarity to the hand-labeled eyelid outline (from 34px to 12px) and  $\approx 21.7$  pixels (or 7.53% of the eye image height) in terms of average eyelid aperture estimation error. The proposed method implementation runs in real time even on a single core ( $\approx$ 7ms) and is available, together with the new data set, at http://www.ti.uni-tuebingen.de/ Eyelid-detection.2007.0.html

## 1. Introduction

The human eye is an important multipurpose organ for which the eyelids serve as protection and maintenance system. Eye protection includes hindering particles from reaching the eye and limiting the amount of light entering the pupil [7]. For maintenance, the eyelids spread the tear film over the cornea during blinks [6]. Thus, eyelashes, blinking, and squinting are essential mechanisms to ensure eye healthiness; however, these mechanisms also create several challenges for computer vision based algorithms that operate on eye images, such as video-based eye tracking. Some of these challenges are shown in Figure 1; for instance, eye features can be occluded by eyelashes or are occluded by the eyelid during a blink (Figure 1a and 1b), eyelid movements can result in motion blur perturbing edge detection (Figure 1c), and false positive detections may arise, such as a wrong pupil detection in the dark circular region on the top right corner of Figure 1d. Due to these challenges, a robust and accurate detection of eyelids can significantly improve the robustness and validity of other methods that are based on the analysis of eye images and videos, such as the automatic detection of eye movements [13, 21] or the analysis of human visual exploration behavior [14, 15]. Moreover, the area restriction imposed by the eyelids allows significant improvements in terms of run time and accuracy of other eye feature detection algorithms (e.g., pupil detection). Furthermore, the eyelid information can be employed for blink detection [2] and, thus, to extract information regarding a person's mental state (e.g., vigilance, fatigue, and drowsiness [17, 26, 22]).



Figure 1: Some of the challenges caused by the eyelids, such as occlusion and motion blur.

In this work, we will focus on single eye images that are produced by head mounted eye trackers (see Figure 2). However, the proposed method makes no assumption regarding the resolution of the images and could be coemployed in images acquired with remote eye trackers to detect eyelids as well as validate the resulting eye-box found by typical face detection algorithms (e.g., Viola-Jones [23]). We introduce a new hand-labeled data set containing 4000 challenging images (Section 3) as well as a novel approach to eyelid identification and aperture estimation (Section 4). The proposed method is evaluated on an existing data set from the literature and on the new data set against two state-of-the-art algorithms, exhibiting performance improvements up to  $\approx 16.11$  percentage points in terms of average similarity to the hand-labeled eyelid outline and  $\approx 21.7$  pixels (or 7.53% of the eye image height) in terms of average eyelid aperture estimation error.



Figure 2: Dikablis essential head mounted eye tracker and an *easy* eye image in terms of computer-vision algorithms; some *hard* instances can be seen in Figure 1.

#### 2. Related Work

Most methods available in the eye-tracking realm focus on the detection of the pupil to derive the gaze information, e.g., [8, 10, 11, 12]. The first eyelid extraction methods originated as a byproduct of attempting to improve iris recognition due to occlusion by the eyelids [5]. Wildes [25] used the Hough transform for parabolas to detect the eyelids. Daugmann [5] first searches for the iris and pupil regions. Within the iris region the upper and lower eyelid are searched as curvilinear edges. Due to the shape change induced to the iris circle by the eyelids, Daugmann [5] uses a statistical spline fitting method for outline estimation. Suzuki et al. [22] starts with partitioning the input image into vertical regions. Afterwards candidates for the upper and lower eyelid are selected based on the intensity distribution of the region. Those candidates are grouped into upper and lower eyelid, and outliers are removed. For eye aperture, the average distance between all vertically corresponding points from both groups is used. Adam et al. [1] first preprocess the input image using anisotropic diffusion, and a Canny edge detector is applied. The region above and below the iris are assume to contain valid eyelid candidates, and edges with a length smaller than the mean edge length are ignored. Parabolic curves are fitted to the remaining candidates for the upper the lower eyelids. For the lower and upper eyelid, the curves with the highest horizontal edge values are selected. Yang et al. [26] define the eyelids as two parabolic curves intersecting at two points, which can be described by four points. To select these points, a likelihood map is generated based on the texture and color similarity areas in the current frame to that of a reference points in a reference frame. Radman et al. [20] perform a radial analysis of the outer iris boundary in the HSI (Hue Saturation Intensity) color space. For each pixel, the saturation of ten adjacent pixels is averaged, and the pixel is considered as an iris-eyelid intersection point if this value exceeds a fixed threshold. This results in two points for the upper and two for the lower eyelid, which are connected using the live wire method [19]. As cost function, a weighted combination of the Canny edge detector, gradient orientation, gradient magnitude, and Laplacian zerocrossing are used. The resulting connection path represents an eyelid. Cai and Wang [4] start with iris and pupil detection. The iris region is filtered using the morphological closing-opening operation to remove eyelashes and punctual bright light spots. Afterwards, the intensity distribution for each column is analyzed. This step is done for the upper and lower part of the iris to seperate between upper and lower eyelid. In each column distribution the minimum is selected as an eyelid point. Afterwards a parabola is fitted to those points in a least squares sense. VASIR [16] is a open source, state-of-the-art iris recognition tool, developed by the National Institute of Standards and Technology. Initially, iris and pupil detections are performed based on circular Hough transforms. Afterwards, linear Hough transforms are used to locate the upper and lower iris regions. The found eyelid points in one region are used to fit a third order polynomial in a least squares sense. The resulting polyniomal is used as eyelid. Fuhl et al. [9] construct a eyelid likelihood map based on a set of features (mean intensity, standard deviation, skewness, and an horizontal Sobel operator); disjointed high likelihood regions are connected using a box filter. Non-maxima surpression and hysteresis are applied, resulting in reconstructed eyelid edges. All resulting edge pairs are evaluated based on features such as enclosed intensity and relative obliqueness. The highest ranked edge pair is selected as upper and lower eyelid reconstructions, and Bezier splines are fit to the corners of these edges to represent the eyelid outline. In this work, we compare the proposed method to the technique proposed by Fuhl et al. [9] since it is the current best performer and runs in real time. Similar to the literature, we also evaluate VASIR [16] as it is a good representative of the methods based on shape recognition.

## 3. Data Sets

As pointed out by Fuhl et al [9], the few publicly available data sets containing eyelid annotations stem from biometric related research, which are typically collected in laboratory conditions and follow guidelines to ensure a certain noise-to-signal ratio; thus, these data are not realistic when real-world scenarios are considered. In their work, Fuhl et al. introduced a realistic data set containing

1100 hand-labeled images, which we employ for evaluation. Furthermore, in this work we are introducing a new publicly available data set containing 4000 hand-labeled eye images, significantly increasing the amount of evaluation data available. Each frame was annotated with ten roughly equally-spaced points: one point on each eye corner (canthus), four points lying on the lower evelid, and four points lying on the upper eyelid (see Figure 3). These images were collected from eleven subjects using a Dikablis Essential head-mounted eye tracker<sup>1</sup> as part of an eyetracking experiment conducted while driving in real-world scenarios [14]. For subjects 1 to 8, we follow the data selection mechanism from [9]: a hundred frames were randomly sampled respecting a minimum temporal distance between frames to increase frame dissimilarity while keeping labeling efforts tractable. Additionally, to evaluate algorithm performance over an extended period of time, for other three subjects we have hand-labeled entire video sections, yielding 1601, 823, and 777, images for subjects 9, 10, and 11, respectively. These data contain high challenging noises for eyelid detection such as strong skin wrinkles, blinks, half blinks, blurry images, reflections, and changing illumination; examples for each subject are shown in Figure 4. The data set and hand-labels can be downloaded at http://www.ti.uni-tuebingen. de/Eyelid-detection.2007.0.html



eye corners eyelid labels eyelid approximation

Figure 3: Labels for eye corners (red) and eyelid points (blue) labels. The eyelid outline can be accurately approximated through Bezier splines fit to the hand-labeled points.

### 4. Method

The general idea behind the proposed approach are oriented edges; in other words, we approximate the upper and bottom eyelids by two functions that define a path along which the sum of orthogonal edge values is maximed. Let e1, e2 be the positions of the eye corners,  $u_{up}(x), b_{bp}(x)$ be the polynomials representing the upper and lower eyelid with parameters up, bp, respectively. The task can then be formulated as the general optimization problem

$$\underset{e_{1,e_{2,up,bp}}}{\operatorname{argmax}} \int_{e_{1}}^{e_{2}} |\Delta u_{up}(x)| + |\Delta b_{bp}(x)| \, dx, \qquad (1)$$

where  $\Delta$  is the difference between the inner and outer intensity values orthogonal to the respective polynomial gradient. Let p be a polynomial line of the form  $p(x) = ax^2 + bx + c$ , then delta is *inner – outer* intensity, where inner/outer depend on the eyelid orientation (lower/upper) along p.

This optimization problem is not convex and, thus, approximations with the Levenberg-Marquardt method can yield wrong maxima [18]. Therefore, either all combinations of e1, e2, up, bp have to be evaluated or a good initialization has to be found. Since the former is prohibitively expensive, we propose a heuristic that tries to approximate this optimization problem by searching for suitable starting positions that restrict the polynomials parameters, thus requiring only partially solving the overall problem. The work flow of the algorithm can be seen in Figure 5, which will be described in detail in the following subsections. It is worth noticing that a downscaling of the input image is ommitted in this figure. This process downscales the input image based on one side of the image, preserving its aspect ratio; this allows us to optimize algorithm parameters and computational costs indepdently of the input image resolution. In our implementation, we employ a median rescaling to an image 76 pixels wide - in practice, this downscales the data sets images by a factor five.

In the first step, the algorithm searches for a possible bottom eyelid location as initial position, assuming the bottom eyelid to be easier to locate. The rationale behind this step is that the bottom eyelid tends to exhibit less variability than its upper counterpart (since its eyelashes are less pronounced) as well as present an overall straighter outline. The position is then validated based on its surrounding intensity; notice that if the eye is slightly open, the upper eyelid assumes a similar form, thus requiring us to analyze if the valid position belongs to the upper or lower eyelid based on its orientation. If the position is valid and correctly oriented, bottom eyelid points are searched through a vertical position optimization paired with an outlier removal mechanism. Polynomials are then iteractively fitted to the left and right side of these points, removing extremeties points when invalid fits are found. The algorithm then approximates the upper eyelid area restriction using the intensity distribution orthogonal to the lower eyelid orientation as well as coarse eye corners locations; from these, the upper eyelid approximation is derived.

#### 4.1. Bottom Eyelid Search

The search of the bottom eyelid is performed solely in the lower two thirds of the input image to avoid eyebrows and reduce computational costs. This step first looks for the maximal row wise summed horizontal edge value (i.e., a straight line) to be used as initial position, based on the bottom eyelid invariability assumption. This values are com-

<sup>&</sup>lt;sup>1</sup>The device produces 384x288 pixels images at 30 frames per second.



Figure 4: Examples from the new data set, which contain 4000 hand-labeled images. Three frames for subjects 1 to 8 (one column per subject) and six frames for subjects 9 to 11 (two columns per subject) are shown.



Figure 5: General overview of the algorithm work flow.

puted as

$$HE_y = \sum_{1}^{X} |I_{x,y-2} + 2I_{x,y-1} - 2I_{x,y+1} + I_{x,y+2}|, \quad (2)$$

where  $I_{x,y}$  is the intensity at position x, y and X the image width. It is worth noticing that a deviation of  $\pm 2$  pixels is employed to compensate for recording skewness and slight bottom eyelid curvature; this approach also translates into less computational costs relative to a employing a vector as indexing orientation. The raw result of this evaluation is shown in green in Figure 6d. However, directly employing these values may yield wrong results due to outliers or recording artifacts, such as the black border at the bottom of Figure 6a, which result in artifical peaks. Hence, the raw signal is smoothed through a mean filter following the range from Equation (2), shown in blue in Figure 6d. The highest peak in the smoothed histogram  $(smooth(HE_y))$  is then selected as starting position (shown as a white line in Figure 6c).

#### 4.2. Bottom Eyelid Validity and Orientation

In particular cases, the previous step may wrongly select the upper eyelid instead of the lower one; for instance, if



Figure 6: The input image (a) and its downscaled version (b). The histogram of horizontally oriented edge values is shown as a green line in (d), whereas the smoothed version is shown as a blue line. Based on the peak in the smoothed histogram, the selected starting position is shown as the white line in (c).

both eyelids are relatively straight (see Figure 7a) or the eye is shut. Thus, we employ the mean intensity above and below the eyelid position for validity and orientation, assuming that the intensity above the lower eyelid must be lower than below it due to the low intensity of iris and dark pupil – the opposite being true for the upper eyelid. If the validity or orientation assumptions are violated, the starting position is then moved to the next  $smooth(HE_y)$  maximum; this procedure is exemplified in Figure 7b and evaluated as

$$Ori(mp_y) = \frac{\sum_{i=1}^{X} \sum_{j=mp_y-1}^{mp_y-10} I_{i,j}}{\sum_{i=1}^{X} \sum_{j=mp_y+1}^{mp_y+10} I_{i,j}},$$
(3)

where  $mp_y$  is the position of the selected eyelid on the y axis and X is the width of the image. If  $Ori(mp_y) > 1$  the next maximum has to be searched; if  $1 \ge Ori(mp_y) > 0.9$  the eye is considered closed.

#### 4.3. Bottom Eyelid Approximation

As can be seen in Figure 7b and Figure 6c, the starting position does not necessarily lays on the lower eyelid. Unfortunately, a complete search over possible lower eye courses is computationally too expensive, and occlusions limit the discovery of pareto-optimal courses. Thus, we ap-



Figure 7: The downscaled input image (a), and an overlay showing the first (and wrong) selected position (white line), which violates the method's assumptions. Thus, the next (and correct) maximum is searched, yielding the correct selection (gray line).



Figure 8: The lower eyelid approximation procedure. The input image (a), starting point (b), and possible candidate points (c). The black dots in (d) are the corrected positions; the remaining points after outliers removal is shown as white dots in (e). The resulting two parabolas are shown in (f).

proach the bottom eyelid approximation through an alternative method. The point lying on the selected  $mp_y$  line (white line in Figure 8b) that maximizes an area difference<sup>2</sup> is selected as starting position (white circle in Figure 8b) to have an entry point for shape evaluation of the later fitted polygons. In practice, this area difference acts as a coarse edge detector, with higher robustness but lower accuracy. Afterwards, candidates for optimization on the line are selected starting from the entry point, using a stride of five pixels, and limited to the inner ninety percent of the image width (white circles in figure 8c). For each candidate, its vertical position is optimized (black dots in figure 8d). This is done by shifting each point vertically until the maximum area difference value is reached (again square of five pixels above and below the actual position). Because the position is already expected to be close to the eyelid and many higher but wrong positions are likely, only continuously increasing maximums are selected in this way. The next step is the first outliers removal step, which is performed bidirectionally and outgoing from the selected starting position. This is done by inspecting the gradient between consecutive optimized candidate positions. The vector  $v = p_2 - p_1$ between two points  $p_1, p_2$  is calculated, which yields the gradient  $g = \frac{v_y}{v_x}$ . Outliers are filtered based on a gradient threshold of one quarter, thus removing consecutive positions with low changes (compare Figure 8c to Figure 8e). In other words, let  $g_1$  and  $g_2$  be consecutive gradients along the polynomial;  $g_2$  is considered an outlier if  $\frac{g_2}{g_1} < \frac{1}{4}$  or  $\frac{g_1}{x} < \frac{1}{4}$ .

 $\frac{g_1}{g_2} < \frac{1}{4}$ . This is a consequence of the five pixel stride between candidates, which should result in more than a single pixel vertical shift. The second outliers removal step is based on the convexity of the resulting least squares polynomial fit<sup>3</sup> (Figure 9). If the resulting polynomial is concave (i.e., curved downwards) (red line in Figure 9) the outermost point is dropped (red dots in Figure 9), and a new least squares polynomial fit is performed. This is done until the resulting polynomial is convex and removes candidates that are out of the range of the eyelid. The resulting two polynomials (i.e., for the left and right directions) are used as bottom eyelid approximation (shown in Figure 8f).



Figure 9: Second outliers removal step illustration. The blue point is the starting point; green and red dots are candidate points. Lines in green have the correct convexity, whereas the red line is concave. Dots in red are removed iteratively until the green line results from the least squares polynomial fit.

#### 4.4. Upper Eyelid Approximation

For the fast approximation of the upper eyelid, first the search region has to be specified, and the coarse positions of upper and bottom eyelid intersections have to be estimated. Therefore, the algorithm starts by selecting a second maximum in  $HE_y$ , but this time over the complete image and starting from the last maximum position. Due to the invalid responses produced by skin folds above the upper eyelid, we search only for the next local maximum. As can be seen in figure 6d, there is another local maximum between both eyelids, which is caused by the pupil and the cornea reflection (white dot below the pupil in figure 6a). The result of

 $<sup>^{2}\</sup>mathrm{This}$  area difference is defined by a square of five pixels above and below the line.

<sup>&</sup>lt;sup>3</sup>In our implementation we used second order polynomials  $ax^2+bx+c$ 

this step can be seen as the white line in figure 10b. The area between this white line and the second gray line in figure 10b is the search region. For future use we define the distance between those lines as  $\Delta WG$ . This second gray line is calculated by doubling the distance between the bottom eyelid and the white line (next local maximum). We do not select the third local maximum as gray line because it is likely that it does not exist.

After the search region has been found, the start, passage and ending points of the upper eyelid polynomial have to be found. In figure 10c the used initial positions are marked as black diamonds. The center diamond is the center of the bottom eyelid on the x axis, around which the algorithm searches for the upper eyelid high position (between the white and the gray line in figure 10b). The left and right black diamonds in figure 10c are the estimated eye corners. Those are set to the position were the bottom eyelid polynomial intersects the starting line of the search region (white line in figure 10b) or to the outmost position on the bottom eyelid. Those initial positions are only coarse, and will be refined in the following three steps.

Due to the computational costs for estimating all three variables at the same time and the non convexity of the problem, we estimate each variable separately. For each variable the algorithmic steps for optimization are:

- 1. Shift position of variable
- 2. Fit polynomial<sup>4</sup> to three points
- 3. Evaluate polynomial with equation 4

$$OEV(f(x)) = \sum_{i=1}^{X} |I(i, f(i) + | \perp f'(i)|) - I(i, f(i) - | \perp f'(i)|)|$$
(4)

Equation 4 describes the valuation of a polynomial based on its oriented edge value. f(x) is the polynomial, I(x, y)the intesity value at location (x, y) and X the width of the image.  $|\perp f'(i)|$  is the normed orthogonal of the tangent of the polynomial at position *i*. OEV(f(x)) is therefore the summed difference between opposite intensity values along the polynomial. In addition to equation 4 we square single differences if the previous difference has the same sign. In other words, if  $d_1$  and  $d_2$  are positive, then  $d_2 = d_2^2$ , where  $d_1$  and  $d_2$  are consecutive deltas along the polynomial.

The first refined position is the high point. Therefore the initial x axis position from the center of the bottom eyelid (centered black diamond in figure 10c) is shifted vertically in the search region and horizontally between  $-\Delta WG$  and  $+\Delta WG$  (step 1 in the enumeration list). The other left and right eye corner stay fixed, and, for each shift of the high

position, a polynomial is fitted to the three points (step 2 in the enumeration list). This polynomials are evaluated with equation 4 (step 3 in the enumeration list) and the maximum is selected. The result of this step can be seen in figure 10d as the gray polynomial.

For the eye corners the same procedure is carried out. The shift region around each initial eye corner position is  $\frac{\Delta WG}{2}$  in each direction. For the left eye corner the result can be seen in figure 10e and the final result in figure 10f.



Figure 10: Upper eyelid approximation procedure. The input images is shown in (a). In (b) the area in which the search takes place is shown through the two lines. The preliminary intersection points and the center are shown as black diamonds in (c). (d) shows the result of the hight approximation (light gray polynomial). In (e) and (f) the approximation for the left and right side of the upper eyelid is shown (white polynomials).

#### 5. Evaluation

As metric for similarity between the estimated evelid outline and the ground truth, we used the Jaccard index, which is typically used for image segmentation evaluation. This index is given by  $\frac{A \cap B}{A \cup B}$ , where A and B are the areas defined by the estimated and ground truth eyelid outlines. Additionally, we evaluated the eyelid aperture estimation error through the Hausdorff distance max(min(d(C, D))), where C and D are the sets of points from the upper and lower eyelid and d() the Euclidean distance (let M be the minimal distance for each point, then the result is max(M)). The eyelid aperture estimation error is then given by the absolute distance between the detected and labeled eyelid Hausdorff distances. We report our results using boxplots: the central mark is the median, edges of the box are the 25th and 75 percentiles, and whiskers extend to the extreme nonoutliers data points; points lying further than  $2.7\sigma$  from the mean are considered outliers and ploted individually<sup>5</sup>.

<sup>&</sup>lt;sup>4</sup>The used polynomial is  $ax^2 + bx + c$ 

 $<sup>{}^5\</sup>sigma$  denotes standard deviation.

Figure 11a shows the overall similarity results – where higher is better. While a score of > 0.5 is already considered similar, the proposed approach reaches a mean similarity score of 0.66 ( $\sigma = 0.18$ ), whereas the method proposed by Fuhl et al. [9] scores 0.5 ( $\sigma = 0.17$ ), and VASIR [16] 0.47 ( $\sigma = 0.24$ ), yielding an improvement of about 16.11 percentage points in terms of average similarity. Regarding the eyelid aperture estimation error (reported in Figure 11b) the proposed method reached an average eyelid aperture estimation error of 12.93 ( $\sigma = 16.72$ ) pixels (or 4.4% of the input image height); in contrast, The method proposed by Fuhl et al. [9] reached 36.84 ( $\sigma = 25.74$ ) and VASIR [16] 34.65 ( $\sigma = 31.66$ ), improving eyelid aperture estimation by  $\approx 21.7$  pixels. Moreover, Fuhl et al. [9] also consider their results in terms of cumulative detection error for an error up to ten pixels, which is shown in Figure 11c; in this regard, the proposed method presents a significant advantage, reaching 61.94% detection rate, against 9.08% by Fuhl et al. [9] and 19.53% by VASIR [16].

Additionally, we report results per subject in Figure 12 to allow for a better accessment on a per-challenge basis, which can be linked to Figure 4. In particular, we call the reader's attention to the last two subjects on the right side (one from the new data set, one from the data set contributed by Fuhl et al. [9]) as these are subjects with glasses, which resulted in several and heavy reflections, making these subjects particularly challenging. Nonetheless, the proposed method is still able to perform satisfactorily, whereas state-of-the-art methods fail. Finally, in terms of average run time, the algorithm by Fuhl et al. [9] was the fastest (3.4ms), with the proposed algorithm requiring 7.1ms, and VASIR [16] 3305.3ms; evaluation was performed using single core C++ implementations on a i5-4570 (@3.2GHz). While not the fastest performer, the proposed method can still reach real-time performance considering state-of-the-art head-mounted eye trackers, which typically require a processing time below 8.33ms (@120 Hz).

## 6. Conclusion

We proposed a real time capable algorithm for eyelid detection. The proposed method outperforms the state-of-the-art in terms of eyelid outline similarity and eyelid aperture estimation. Furthermore, we significantly increase the amount of publicly available handlabeled data for eyelid detection evaluation by providing a new data set with over 4000 images from realworld and challenging scenarios. To foster further research in the area, we provide the data set and C++ code freely at http://www.ti.uni-tuebingen. de/Eyelid-detection.2007.0.html

Future work will include the evaluation on remote eye tracking data and also algorithmic improvements concerning the use of regression forests for an supportive candidate selec-



Figure 11: Overall results in terms of outline similarity, eyelid aperture estimation, and cumulative detection rate.

tion, as proposed in [24]. In addition the use of a parametric eye model [3] would help validating results and reconstruct covered eyelids.

#### References

- M. Adam, F. Rossant, F. Amiel, B. Mikovikova, and T. Ea. Eyelid localization for iris identification. *Radioengineering*, 17(4):82–85, 2008.
- [2] T. Appel, T. Santini, and E. Kasneci. Brightness-and motionbased blink detection for head-mounted eye trackers. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, pages 1726– 1735. ACM, 2016.
- [3] P. Bérard, D. Bradley, M. Gross, and T. Beeler. Lightweight eye capture using a parametric model. *TOG*, 35, 2016.
- [4] P. Cai and C. Wang. An eyelid detection algorithm for the iris recognition. *IJSIA*, 9(5):105–112, 2015.
- [5] J. Daugman. How iris recognition works. *TCSVT*, 14(1):21– 30, 2004.



Figure 12: Outline similarity (top) and eyelid aperture estimation error (bottom) per subject.

- [6] P. Dilly. Structure and function of the tear film. In *Lacrimal Gland, Tear Film, and Dry Eye Syndromes*, pages 239–247. Springer, 1994.
- [7] F. Fralick. Anatomy and physiology of the eyelid. *Trans Am Acad Ophthalmol Otolaryngol*, 66:575, 1962.
- [8] W. Fuhl, T. Kübler, K. Sippel, W. Rosenstiel, and E. Kasneci. Excuse: Robust pupil detection in real-world scenarios. In *CAIP*, pages 39–51. Springer, 2015.
- [9] W. Fuhl, T. Santini, D. Geisler, T. Kübler, W. Rosenstiel, and E. Kasneci. Eyes wide open? eyelid location and eye aperture estimation for pervasive eye tracking in real-world scenarios. In *PETMEI*, 09 2016.
- [10] W. Fuhl, T. Santini, G. Kasneci, and E. Kasneci. Pupil-

net: Convolutional neural networks for robust pupil detection. arXiv preprint arXiv:1601.04902, 2016.

- [11] W. Fuhl, T. C. Santini, T. Kübler, and E. Kasneci. Else: Ellipse selection for robust pupil detection in real-world environments. In *ETRA*, pages 123–130, New York, NY, USA, 2016. ACM.
- [12] W. Fuhl, M. Tonsen, A. Bulling, and E. Kasneci. Pupil detection for head-mounted eye tracking in the wild: an evaluation of the state of the art. *Machine Vision and Applications*, pages 1–14, 2016.
- [13] E. Kasneci, G. Kasneci, T. C. Kübler, and W. Rosenstiel. Online recognition of fixations, saccades, and smooth pursuits for automated analysis of traffic hazard perception. Ar-

tificial Neural Networks: Methods and Applications in Bio-/Neuroinformatics, pages "411–434", 2015.

- [14] E. Kasneci, K. Sippel, K. Aehling, M. Heister, W. Rosenstiel, U. Schiefer, and E. Papageorgiou. 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.
- [15] T. C. Kübler, E. Kasneci, and W. Rosenstiel. Subsmatch: Scanpath similarity in dynamic scenes based on subsequence frequencies. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 319–322. ACM, 2014.
- [16] Y. Lee, R. J. Micheals, J. J. Filliben, and P. J. Phillips. Vasir: an open-source research platform for advanced iris recognition technologies. *Journal of Research of NIST*, 118:218, 2013.
- [17] L. K. McIntire, R. A. McKinley, C. Goodyear, and J. P. McIntire. Detection of vigilance performance using eye blinks. *Applied ergonomics*, 45(2):354–362, 2014.
- [18] J. J. Moré. The levenberg-marquardt algorithm: implementation and theory. In *Numerical analysis*, pages 105–116. Springer, 1978.
- [19] E. Mortensen, B. Morse, W. Barrett, and J. Udupa. Adaptive boundary detection using 'live-wire' two-dimensional dynamic programming. IEEE, 1992.
- [20] A. Radman, N. Zainal, and M. Ismail. Efficient iris segmentation based on eyelid detection. *Journal of Engineering Science and Technology*, 8(4):399–405, 2013.
- [21] T. Santini, W. Fuhl, T. Kübler, and E. Kasneci. Bayesian identification of fixations, saccades, and smooth pursuits. In *ETRA*, pages 163–170, New York, NY, USA, 2016. ACM.
- [22] M. Suzuki, N. Yamamoto, O. Yamamoto, T. Nakano, and S. Yamamoto. Measurement of driver's consciousness by image processing-a method for presuming driver's drowsiness by eye-blinks coping with individual differences. In *SMC*, volume 4, pages 2891–2896. IEEE, 2006.
- [23] M. Viola, M. J. Jones, and P. Viola. Fast multi-view face detection. In CVPR. Citeseer, 2003.
- [24] C. Wang, F. Shi, S. Xia, and J. Chai. Realtime 3d eye gaze animation using a single rgb camera. *TOG*, 35(4):118, 2016.
- [25] R. P. Wildes. Iris recognition: an emerging biometric technology. *Proceedings of the IEEE*, 85(9):1348–1363, 1997.
- [26] F. Yang, X. Yu, J. Huang, P. Yang, and D. Metaxas. Robust eyelid tracking for fatigue detection. In *ICIP*, pages 1829– 1832, Sept 2012.