

Pupil detection in the wild: An evaluation of the state of the art in mobile head-mounted eye tracking

Wolfgang Fuhl · Marc Tonsen · Andreas Bulling ·
Enkelejda Kasneci

Preprint

Abstract Robust and accurate detection of the pupil position is a key building block for head-mounted eye tracking and prerequisite for applications on top, such as gaze-based human-computer interaction or attention analysis. Despite a large body of work, detecting the pupil in images recorded under real-world conditions is challenging given significant variability in the eye appearance (e.g., illumination, reflections, occlusions, etc.), individual differences in eye physiology, as well as other sources of noise, such as contact lenses or make-up. In this paper we review six state-of-the-art pupil detection methods, namely ElSe [6], ExCuSe [5], Pupil Labs [16], SET [8], Starburst [18], and

Świrski [30]. We compare their performance on a large-scale dataset consisting of 225,569 annotated eye images taken from four publicly available datasets. Our experimental results show that the algorithm ElSe [6] outperforms other pupil detection methods by a large margin, offering thus robust and accurate pupil positions on challenging everyday eye images.

Keywords Pupil Detection · Head-Mounted Eye Tracking · Data set · Computer Vision · Image processing

1 Introduction

Understanding processes underlying visual perception has been a focus in various research fields including medicine, psychology, advertisement, autonomous driving, or application control. Recent developments in head-mounted eye-tracking have enabled researchers to study human visual perception and attention allocation in natural environments. Such systems generally fall into two categories, i.e., remote tracking systems, where the subject is recorded with an external camera, and head-mounted eye-trackers. The main challenges that have to be faced in remote eye tracking are the robust detection of the subject's face and eyes. Several techniques have been proposed for the robust

W. Fuhl
Perception Engineering Group, University of
Tübingen, Germany
E-mail: wolfgang.fuhl@uni-tuebingen.de

M. Tonsen
Perceptual User Interfaces Group, Max Planck In-
stitute for Informatics, Germany
E-mail: tonsen@mpi-inf.mpg.de

A. Bulling
Perceptual User Interfaces Group, Max Planck In-
stitute for Informatics, Germany
E-mail: bulling@mpi-inf.mpg.de

E. Kasneci
Perception Engineering Group, University of
Tübingen, Germany
E-mail: Enkelejda.Kasneci@uni-tuebingen.de

face (e.g. [9, 10]) and eye recognition as well as hallucination (e.g., [11]) in low resolution images.

The measurement of eye movements in head-mounted eye-trackers, however, is based on one camera that records the viewed scene and at least one additional camera, which is directed towards the subject’s eye to record the eye movements (e.g., Dikablis Mobile eye tracker, Pupil Labs eye tracker [16], SMI Glasses, Tobii Glasses). The gaze point is then mapped to the viewed scene based on the center of the pupil and a user-specific calibration routine. A crucial prerequisite for a robust tracking is an accurate detection of the pupil center in the eye images. While eye tracking can be accomplished successfully under laboratory conditions, many studies report the occurrence of difficulties when eye trackers are employed in natural environments, such as driving [1, 12, 15, 20], shopping [14, 26], or simply walking around [28].

The main source of error in such settings is a non-robust pupil signal that is mainly related to challenges in the image-based detection of the pupil. Schnipke and Todd summarized in [25] a variety of difficulties occurring when using eye trackers, such as changing illumination, motion blur, recording errors, and eyelashes covering the pupil. Rapidly changing illumination conditions arise primarily in tasks where the subject is moving fast (e.g., while driving) or rotates relative to unequally distributed light sources. Furthermore, in case the subject is wearing eye glasses or contact lenses, further reflections may occur (see Figure 7, Data set III and XVIII). A further issue arises due to the off-axial position of eye camera in head-mounted eye trackers, e.g., when the pupil is surrounded by a dark region (see Figure 7, Data set VII). Other difficulties are often posed by physiological eye characteristics, which may interfere with detection algorithms, such as additional dark spots on the iris. Therefore, studies based on eye tracking in uncontrolled environments constantly report low pupil detection rates [13, 20, 33]. As a consequence, the data collected in such studies has to be

manually post-processed, which is a laborious and time-consuming procedure. Furthermore, such post-processing is impossible for real-time applications that rely on pupil monitoring (e.g., driving assistance based on eye-tracking input [31], gaze-based interaction [22, 27, 34], eye-based activity and context recognition [1, 2, 3] and many more). In light of the above challenges, real-time and accurate pupil detection is an essential prerequisite for pervasive video-based eye-tracking.

2 State-of-the-art methods for pupil detection

Over the last years, there has been a large body of work on pupil detection. However, most of the approaches address pupil detection under laboratory conditions, e.g., in [7, 17], both employing a histogram-based threshold. Such algorithms can be applied to eye images captured under infrared light as in [19], [21], or [24]. Another threshold-based approach was presented in [38], where the pupil is detected based on the calculation of the curvature of the threshold border. An isophotes curvature-based approach is presented in [35], using the maximum isocenter as pupil center estimation. Probably the most popular algorithm in this realm is Starburst, introduced by [18] in 2005. In 2012, Swirski et. al [30] proposed an algorithm, which starts with a coarse positioning using Haar-like features and then refines the pupil center position.

The open source head-mounted eye tracker from Pupil Labs also comes with a pupil detection algorithm designed for unconstrained everyday settings [16]. Wood et al. presented a model-based gaze estimation system for unmodified tablet computers [37]. Their gaze estimation pipeline also includes a Haar-like feature based eye detector as well as a RANSAC-based limbus ellipse fitting approach. Three recent methods, SET [8], ExCuSe [5], and ElSe [6], explicitly address the aforementioned challenges

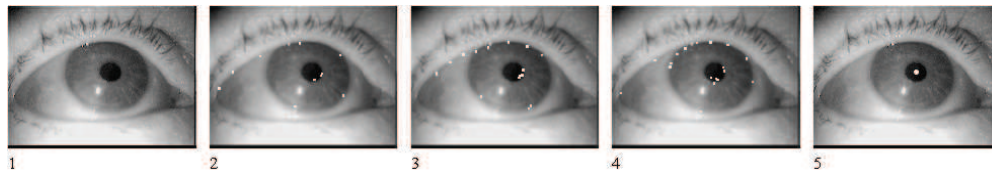


Fig. 1 Visualization of the Starburst [18] algorithm. First the input image (1) is smoothed. Rays are sent out from a starting position, where edge values exceeding an edge threshold are selected as pupil contour candidates (2). These points serve then as new starting points for rays going in the opposite direction, providing thus new contour candidates (3). The search process is repeated iteratively until convergence (4). The pupil center is finally estimated using a RANSAC ellipse fit (5).

associated with pupil detection in natural environments. SET is based on thresholding and ellipse fitting. ExCuSe [5] first analyses the input images with regard to reflections based on intensity histograms. Several processing steps based on edge detectors, morphologic operations, and the Angular Integral Projection Function are then applied to extract the pupil contour. Similar to ExCuSe, ElSe is also based on edge filtering, ellipse evaluation, and pupil contour validation [6].

The algorithms Starburst [18], Świrski [30], Pupil Labs [16], SET [8], ExCuSe [5], ElSe [6] will be presented and discussed in detail in the following subsections. We compared these algorithms on a large corpus of hand-labeled eye images (Section 3) and will present the results in Section 4.

2.1 Starburst

In the first step of Starburst [18] the image is denoised using a Gaussian filter. The algorithm then uses adaptive thresholding on a square region of interest in each video frame to localize the corneal reflection. The location of the corneal reflection is given by the geometric center of the largest region in the image using the adaptively determined threshold. Radial interpolation is further used to remove the corneal reflection from the eye image. The central step of the algorithm that also gave this method its name, is to estimate the pupil contour by detecting edges along a limited number of rays that extend from a central best guess of the pupil center (see Figure 1.2,3,4). The

rays are independently evaluated pixel by pixel until a threshold is exceeded, indicating the edge of the pupil. A feature point is defined at that location as a contour edge candidate and the processing along the ray is stopped. For each pupil candidate, another set of rays is generated that creates a second set of pupil contour candidates. This process is iterated until convergence. Model fitting is finally performed following a Random Sample Consensus (RANSAC) paradigm to find the best fitting ellipse describing the pupil boundary. This result is further improved through a model-based optimization that does not rely on feature detection [18].

2.2 Świrski

The algorithm by Świrski et al. [30] works in three main stages (see Figure 3). To find the pupil, the Świrski detector first calculates the integral image and convolves it with a Haar-like feature, similar to the features used in cascade classifiers [36]. The algorithm repeats this convolution for a set of possible radii between a user-specified minimum and maximum.

The position of the strongest response is the estimated center of the pupil, with the size of the region determined by the corresponding radius. The initial region estimation is unlikely to be accurately centered on the pupil. Therefore, in the second step, the algorithm approximates the pupil position within this region based on K-means clustering: the image histogram is segmented in dark (pupil pixels) and light. The



Fig. 2 Visualization of the SET algorithm. (1) input image and (2) result of binarization. (3) image after applying a region size threshold to extract pupil candidates. (4) candidate pupil centers and (5) final result.

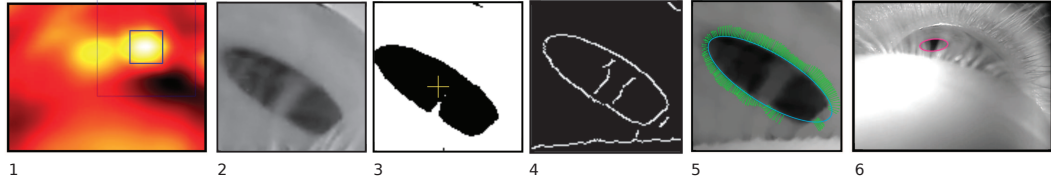


Fig. 3 Visualization of the Świrski algorithm [30]. First, the eye image is convolved with a Haar-like center surround feature and the location of the maximal response (1) is used as the pupil region (2). The pupil region is segmented using k-means clustering of its histogram (3) and passed through a Canny edge detector (4). The algorithm finally uses an image-aware Random Sample Consensus (RANSAC) ellipse fitting (5) to detect the pupil (6).

algorithm then creates a segmented binary image of the pupil region by thresholding any pixels above the maximum intensity in the dark cluster. Afterwards, connected components in the segmented image are found and the largest among these is selected to be the pupil.

The center of mass of this component approximates the pupil position. The final stage of the algorithm refines the pupil position estimate by fitting an ellipse to the boundary between the pupil and the iris. To do this, the image is preprocessed to create an edge image and to robustly fit an ellipse to the edge points while ignoring any outliers. To remove features such as eyelashes and glints, a morphological “open” operation is performed to close small bright gaps in the otherwise dark pupil region, without significantly affecting the pupil’s contour. The algorithm then finds the boundary between pupil and iris using a Canny edge detector. Finally, the algorithm fits an ellipse to the edge pixels using RANSAC as well as an image-aware support function. The support function ensures that the ellipse lies on a boundary from dark pixels to light pixels, and that they lie along strong image edges.

2.3 Pupil Labs

The Pupil Labs detector is integrated into the open source head-mounted eye tracking platform Pupil [16]. Figure 4 visualizes the different processing steps of the algorithm. In a first step, the eye image is converted to grayscale. The initial region estimation of the pupil is found via the strongest response for a center-surround feature as proposed in [30]. The algorithm then uses a Canny filter to detect edges in the eye image and filters these edges based on neighboring pixel intensity. It then looks for darker areas (blue region) while dark is specified using a user set offset of the lowest peak in the histogram of pixel intensities in the eye image. Remaining edges are filtered to exclude those stemming from spectral reflections (yellow region) and extracted into contours using connected components [29]. The contours are simplified using the Ramer-Douglas-Peucker algorithm [4] and then filtered and split into sub-contours based on criteria of curvature continuity. Candidate pupil ellipses are found using ellipse fitting onto a subset of the contours. Good fits are defined in a least square sense, major radii within a user defined range and ellipse center in a “blue

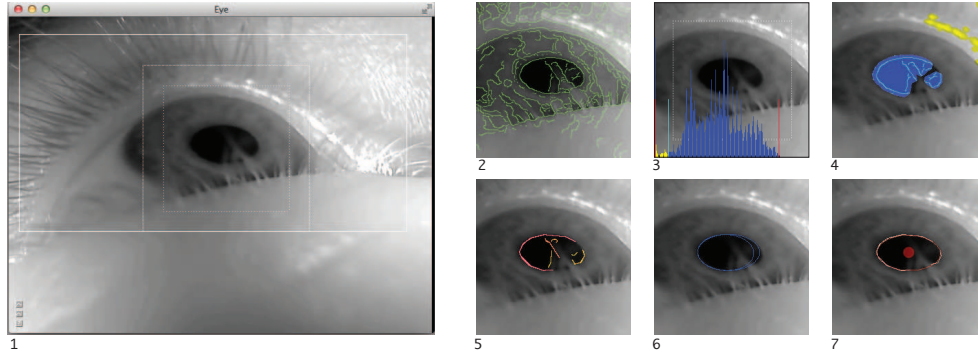


Fig. 4 Visualization of the Pupil Labs algorithm [16]. 1) Eye image, user region of interest (white stroke rectangle), and initial estimation of pupil region (white square and dashed line square.) 2) Canny edge detection (green lines.) 3) Define “dark” region as offset from lowest spike in histogram within eye image. 4) Filter edges to exclude spectral reflections (yellow.) and not inside “dark” areas (blue) 5) Remaining edges extracted into contours using connected components and split into sub-contours based on curvature continuity criteria (multi colored lines). 6) Candidate pupil ellipses (blue) are formed using ellipse fitting 7) Final ellipse fit found through an augmented combinatorial search (finally ellipse with center in red) - supporting edge pixels drawn in white.

region” (see Figure 4.4). An augmented combinatorial search looks for contours that can be added as support to the candidate ellipses. The results are evaluated based on the ellipse fit of the supporting edges and the ratio of supporting edge length and ellipse circumference. If the ratio is above a threshold the algorithm uses the raw canny edge pixels to refine the ellipse fit and reports this final ellipse as the one defining the contour of the pupil. Otherwise the algorithm reports that no pupil was found.

2.4 SET

The SET approach consists of a combination of manual and automatic step estimate the pupil center [8]. Prior to pupil detection, two parameters, i.e., a threshold to convert the input to a binary image (see Figure 2(2)) and the size of the segments (see Figure 2.3) considered for pupil detection are set manually [8]. The thresholded image is first segmented and pixels are then grouped into maximal connected regions to find pupil candidates [8]. For each segment larger than a threshold value, the Convex Hull method is applied to compute the segment border. In the last step, an ellipse is fitted to each

extracted segment; (see Figure 2(4)). The ellipse that is closest to a circle is considered as the pupil (see Figure 2.5).

2.5 ExCuSe

ExCuSe is a recently introduced algorithm that builds on edge detection and morphologic operations [5]. The algorithmic work-flow is presented in Figure 5. In the first step, the input image is normalized and a histogram of the image is calculated. If a peak in the bright histogram area is found, the pupil can be found based on edge analysis (first row in Figure 5). To achieve this, a Canny edge detector is applied to the input image (see Figure 5(2)). The resulting edges are then refined by removing thin lines and thinning thick edges using a morphologic operator. All remaining edges are smoothed and orthogonal edges are removed using morphologic patterns (see Figure 5(3)). For each connected line, the mean position is calculated. Based on this information, straight lines are excluded from further processing. All remaining curved lines are kept as shown in Figure 5(4) and further processed. For each remaining curve, the enclosed mean intensity

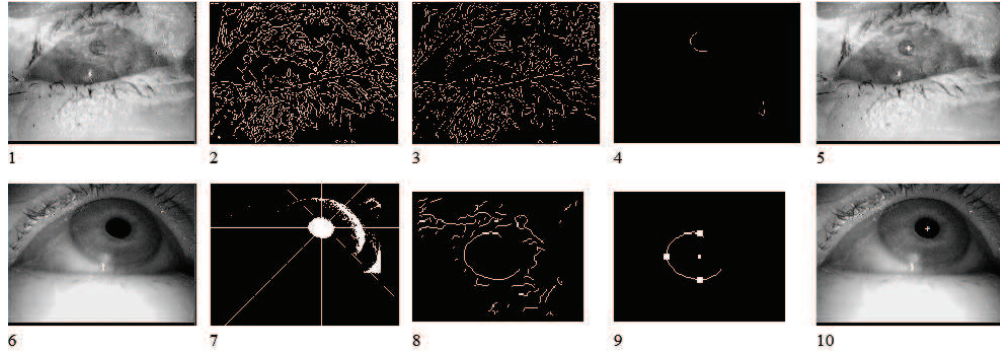


Fig. 5 The algorithmic work-flow in ExCuSe [5]. (1) input image with reflections and (2) Canny edge filtered image. (3) refined edges using morphologic operators. Remaining edges are analyzed regarding their curvature (4). The best edge is fitted by an ellipse and its center is reported as the pupil center (5). (6) input image without reflections. (7) Coarse pupil estimate based on the angular integral projection. (8) Pupil position refinement based on the Canny edge image. (9) rays are send out of the optimized position to select edges representing the pupil border (white dots on line in 9 are ray hits). (10) result of the pupil center estimation.

value is calculated and the curve with the lowest value is chosen as pupil curve. Afterwards, an ellipse is fitted to this curve (see Figure 5(5)).

In case no bright peak in the intensity histogram is detected, a threshold based on the standard deviation of the image is calculated. This corresponding steps are visualized in the second row of Figure 5. First the algorithm determines a coarse pupil position and then refines this stepwise to approach the pupil center. The coarse pupil position is estimated based on the Angular Integral Projection Function (AIPF) [23] on the thresholded image. More specifically, the input image is thresholded and pixels over the threshold are summed along the rows. This summation is done four times by rotating with an orientation angle of 45° , Step 7 in Figure 5. Once a coarse pupil center estimation has been performed, a refinement step based on features of the surrounding neighborhood is performed, Step 8 in Figure 5. The assumption here is that pixels belonging to the pupil are surrounded by brighter or equally bright pixels. Finally, a thresholded image is used in ExCuSe to improve the edge image and refine the pupil edges. The result for the input image from Figure 5(6) is shown in Figure 5(10).

2.6 ElSe

Similar to ExCuSe, the ElSe algorithm operates on Canny-edge filtered eye images. The pupil center is found in a decision-based approach as described in [6]. Based on the edge-filtered image, edge connections that could impair the surrounding edge of the pupil are removed by means of morphologic operators (see Figure 6(2)). Afterwards, connected edges are collected and evaluated based on straightness, inner intensity value, elliptic properties, the possibility to fit an ellipse to it, and a pupil plausibility check (see Figure 6(3)). If a valid ellipse describing the pupil is found, it is returned as the result (see Figure 6(4)).

In case no ellipse is found (e.g., when the edge filtering does not result in suitable edges), a second analysis is conducted. More specifically, as in ExCuSe, ElSe first estimates a likely location candidate and then refines this position. Since a computationally demanding convolution operation is required, the image is rescaled to keep run-time tractable. This rescaling process contains a low pass procedure to preserve dark regions (see Figure 6(7)) and to reduce the effect of blurring or noise caused by eyelashes in the eye image. Afterwards, the image is convolved with two different filters sep-

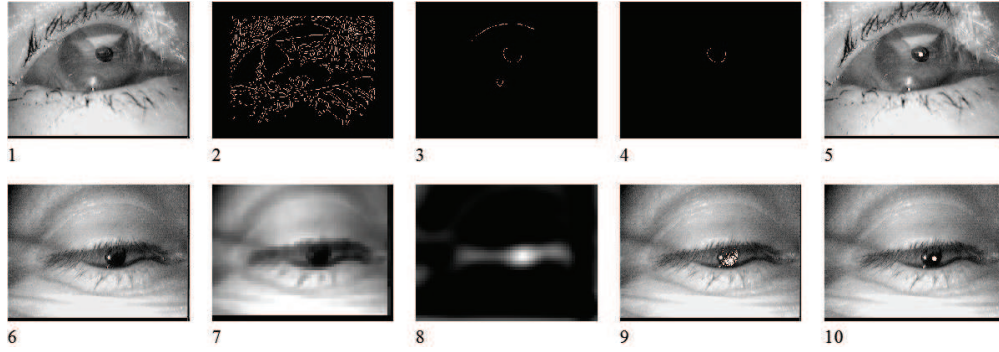


Fig. 6 The algorithmic work-flow in ElSe [6]. (1) input image, (2) Canny edge image after morphologic operations. (3) remaining edges after curvature analysis, analysis of the enclosed intensity value, and shape. (4) the curvature with the most enclosing low intensity values and the most circular ellipse is chosen as the pupil boundary. (5) pupil center result. If ElSe [6] fails to select a valid ellipse, the input image (6) is downscaled (7). (8) image after convolution with a mean and a surface difference filter. (9) a threshold for pupil region extraction is calculated and used for pupil area estimation. (10) pupil center result.

arately: 1) a surface difference filter to calculate the area difference between an inner circle and a surrounding box, and 2) a mean filter. The results of both convolutions are multiplied (see Figure 6(8)), and the maximum value is set as the starting point of the refinement step. Since choosing a pixel position in the downscaled image leads to a distance error of the pupil center in the full scale image, the position has to be optimized on the full scale image based on an analysis of the surrounding pixels of the chosen position (see Figure 6(9)). The center of mass of the pixels under this threshold is used as the new pupil position (see Figure 6(10)). This position is evaluated regarding the possibility to be the pupil by analyzing the surface difference result of a validity pattern.

3 Data sets

3.1 Świrski

The dataset introduced by Świrski et al. [30] in 2012 provides 600 manually labeled, high-resolution (640x480 pixels) eye images. The data was collected during in-door experiments with 2 subjects and 4 different camera angles. The main challenges in pupil detection arise due to highly off-axial camera position and occlusion of the pupil by the eye lid.

3.2 ExCuSe

This data set was recently provided with the ExCuSe [5] algorithm and includes 38,401 high-quality, manually labeled eye images (384x288 pixels) from 17 different subjects¹. Exemplary images are shown in Figure 7. The first nine data sets in ExCuSe were recorded during an on-road driving experiment [13] using a head-mounted Dikablis Mobile eye tracker. The remaining eight data sets were recorded during a supermarket search task [26]. These data sets are highly challenging, since illumination conditions change frequently. Furthermore, reflections on eye glasses and contact lenses often occur (Table 1). The experiments were not specifically designed to pose challenges to pupil detection, but reflect typical data collected out of the laboratory.

3.3 ElSe

The Data sets XVIII-XXIV (Table 1 and Figure 7) were recently published with the ElSe algorithm in [6]. This data set contains overall 55,712 eye images (384x288 pixels) collected from seven subjects wearing a Dikablis eye

¹ <https://www.ti.uni-tuebingen.de/Pupil-detection.1827.0.html?&L=1>

Data set	Eye images	Description
Świrski [30]	600	Highly off-axis, pupil occluded by eye lashes
ExCuSe [5]	I	Reflections
	II	Reflections, changing illumination conditions
	III	Reflections, recording errors, bad illumination conditions
	IV	Contact lenses, bad illumination
	V	Shifted contact lenses
	VI	Bad illumination, mascara
	VII	Bad illumination, mascara, eyeshadow
	VIII	Bad illumination, pupil occluded by eyelashes
	IX	Reflections, additional black dot on iris
	X	Bad illumination, highly of-axis (pupil at image boarder)
	XI	Reflections, bad illumination, additional black dot on iris
	XII	Bad illumination
	XIII	Bad illumination, eyelashes covering pupil
	XIV	Bad illumination
	XV	Shifted contact lenses
	XVI	Mascara, eyelashes
	XVII	Bad illumination, eyelashes covering pupil
ElSe [6]	XVIII	Reflections, changing illumination conditions
	XIX	Reflections
	XX	Reflections, changing illumination conditions
	XXI	Bad illumination, Reflections
	XXII	Reflections, changing illumination conditions
	XXIII	Asian subject
	XXIV	Asian subject, reflections
LPW [32]	XXV	Changing illumination conditions, eyelashes covering pupil
	XXVI	Bad illumination
	XXVII	Reflections, changing illumination conditions
	XXVIII	Bad illumination, off-axis, reflections, pupil at border
	XXIX	Border of glasses covering pupil, blurred images
	XXX	Eyelashes covering pupil, pupil at border
	XXXI	Reflections, small pupil, bad illumination, pupil at border
	XXXII	Reflections, small pupil, bad illumination, pupil at border
	XXXIII	Eyelashes covering pupil, pupil at border
	XXXIV	Highly off-axis, bad illumination, pupil at border
	XXXV	Eyelashes covering pupil, pupil at border
	XXXVI	Bad illumination, pupil at border
	XXXVII	Reflections, bad illumination, eyelashes covering pupil
	XXXVIII	Highly off-axis, eyelashes covering pupil
	XXXIX	Reflections
	XXXX	Reflections, bad illumination
	XXXXI	Highly off-axis, eyelashes covering pupil, bad illumination
	XXXXII	Reflections, bad illumination
	XXXXIII	Highly off-axis, pupil at border, upside down, reflections
	XXXXIV	Bad illumination, pupil at border
	XXXXV	Pupil at image border
	XXXXVI	Mascara, pupil at border

Table 1 Four publicly available data sets containing 225,569 ground-truth eye images were employed for the evaluation of pupil detection algorithms. Explicit challenges associated with each data set are mentioned shortly.

tracker during various tasks. The Data sets XVIII-XXII were derived from eye-tracking recordings during an on-road driving experiment [13]. The remaining Data sets XXIII and XXIV were recorded during in-door experiments with two Asian subjects. The challenge

in pupil detection arises from eyelids and eyelashes occluding the pupil or casting shadows onto the pupil (and, in one case, glasses reflections). Further challenges associated with Data set XXIV are related to reflections on eye glasses. The challenges in the eye images

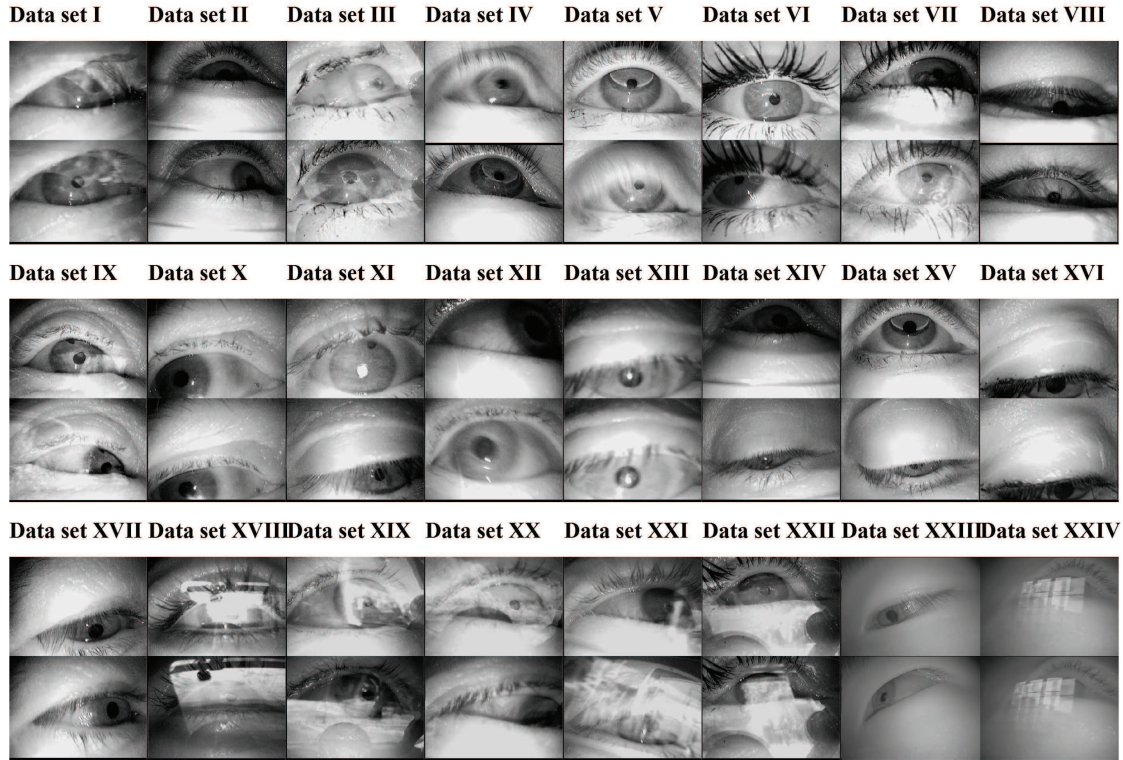


Fig. 7 Exemplary eye images of each data set introduced together with the pupil detection algorithms ExCuSe [5] and ElSe [6]. Data sets I - IX and XVIII - XXII were collected during an on-road driving experiment with different subjects. The Data sets X - XVII were recorded during a supermarket search task. The remaining Data sets XXIII and XXIV were recorded in a laboratory setting with two Asian subjects.

included in the Data sets XVIII, XIX, XX, XXI and XXII are related to motion blur, reflections, and low pupil contrast in comparison with to the surrounding area.

3.4 Labeled pupils in the wild (LPW)

The recent Labeled Pupils in the Wild (LPW) dataset [32] contains 66 high-quality eye region videos that were recorded from 22 participants using a dark-pupil head-mounted eye tracker from Pupil Labs [16]. Each video in the dataset consists of about 2,000 frames with a resolution of 640x480 pixels and was recorded at about 95 FPS, resulting in a total of 130,856 video frames. The dataset is one order of magnitude larger than other datasets and covers a wide range of realistic indoor and outdoor illumination conditions, includes participants wearing

glasses and eye make-up, as well as different ethnicities with variable skin tones, eye colors, and face shapes (see Figure 8). All videos were manually ground-truth annotated with accurate pupil center positions.

4 Experimental Results

We compared the algorithms Starburst² [18], SET³ [8], Świrski et al.⁴ [30], PupilLabs⁵ [16],

² <http://thirtysixthspan.com/openEyes/software.html>

³ <https://sites.google.com/site/eyegoeyetracker/>

⁴ <https://www.cl.cam.ac.uk/research/rainbow/projects/pupiltracking/>

⁵ <https://github.com/pupil-labs/pupil>

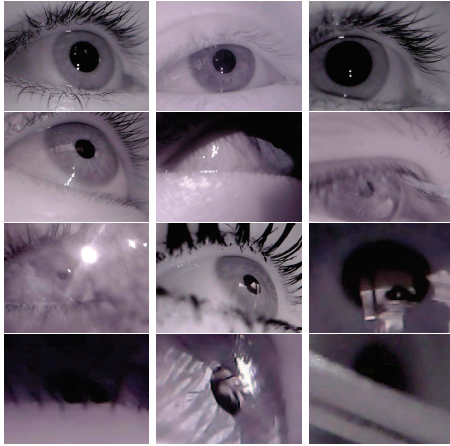


Fig. 8 Example images of the labeled pupils in the wild (LPW) dataset. The top row shows different eye appearances. The second row shows particularly difficult cases, such as strong shadows, occlusions from eye lids, reflections on glasses, and make-up. The third row shows close-ups to illustrate the effect of reflections, self occlusions, strong sunlight and shadow, as well as occlusions by glasses.

ExCuSe⁶ [5], and ElSe⁶ [6] on the above data sets from Table 1. All algorithms were employed with their default parameter setting. We report the performance of the above algorithms in terms of the detection rate for different pixel errors, where the pixel error represents the Euclidean distance between the manually labeled center of the pupil and the pupil center reported by the algorithm. Note that we do not report performance measures related to the gaze position on the scene, since this also depends on the calibration. We focus on the pupil center position on the eye images, where the first source of noise occurs.

Table 2 summarizes the performance of the evaluated algorithms on each data set. On 42 out of 47 data sets, ElSe [6] clearly outperformed the other state-of-the-art algorithms, being thus the most promising approach towards robust pupil detection in heavily noisy eye images. The average detection rates of the evaluated algorithms on the whole image corpus (i.e., 225,569 ground-truth eye images from

Table 1) are presented in Figure 9. Note that the results are weighted by the number of images on each data set. As shown in the Figure, ElSe shows superior performance, reaching an average detection rate of more than 60% at a pixel distance error of 5.

A detailed performance analyses on each data set is visualized in Figure 10. The highest detection rates are achieved on the Świrski et al. [30] data set. Since this data set was collected in a laboratory setting, it is the least challenging, although most of the contained eye images are highly off-axial. For this data set, the algorithms ExCuSe, ElSe, and Swirski reach a detection rate far beyond 70% at a pixel distance of 5. With a detection rate of 86.17% (Table 2), ExCuSe is the best performing algorithm among the state-of-the-art.

The data sets ExCuSe, ElSe, and LPW provide a large corpus of eye images collected in outdoor scenarios and represent the various challenges that have to be faced when head-mounted eye trackers are employed in such settings. Figure 10(b) shows the evaluation result on the ExCuSe data set. For this data set, ElSe is the best performing algorithm with a detection rate of 70% at a pixel error of 5. The ExCuSe algorithms achieves also good detection rates of about 55%, whereas the remaining algorithms show detection rates below 30%.

Due to the many sources of noise summarized in Table 1, the ElSe data set contains the most challenging eye images. The best detection rates (for a pixel error of 5) on this data set are achieved by the algorithms ElSe (50%) and ExCuSe (35%), while the remaining algorithms show detection rates of at most 10%. According to the evaluation results on the LPW data set (Figure 10(d)), ElSe proves to be the most robust algorithm when employed in outdoor scenarios. At a pixel error of 5, ElSe shows a detection rate of 70%. Good detection rates (50%) are also achieved by the algorithms ExCuSe and Swirski, whereas the remaining approaches have detection rates below 40%.

⁶ <https://www.ti.uni-tuebingen.de/Pupil-detection.1827.0.html?&L=1>

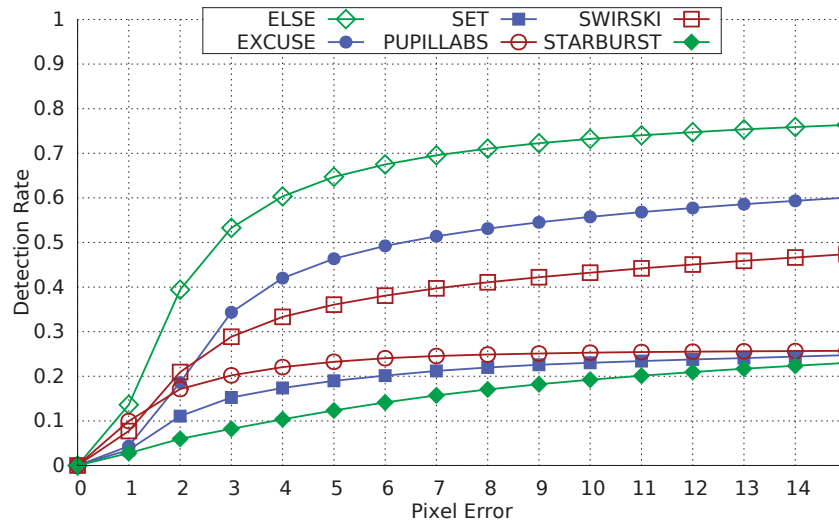
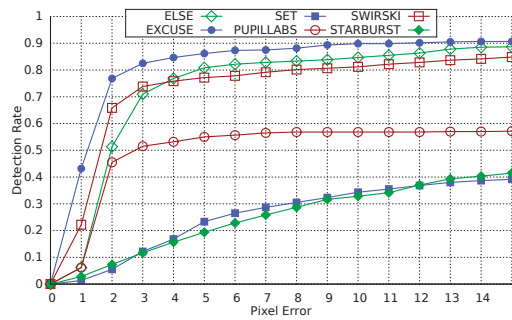
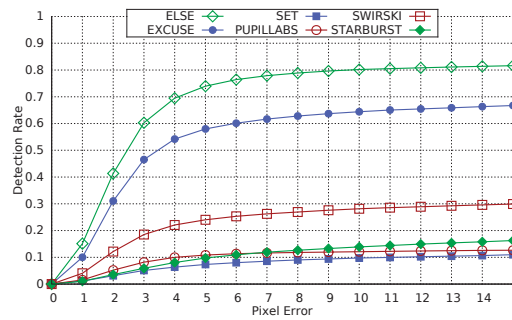


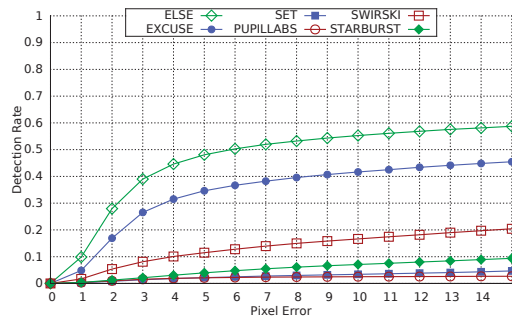
Fig. 9 Average detection rates at different pixel distances for all data sets. The result for each data set is weighted by the number of images in the corresponding data set.



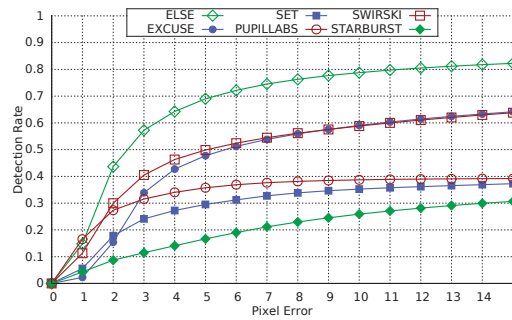
(a) Performance on the Świrski data set



(b) Performance on the ExCuSe data set



(c) Performance on the ElSe data set



(d) Performance on the LPW data set

Fig. 10 Detection rates of the algorithms ElSe, ExCuSe, Pupil Labs, SET, Starburst, and Świrski for each of the data sets described in Table 1.

Data set	SET(%)	Starburst(%)	Świrski(%)	ExCuSe(%)	ElSe(%)	PupilLabs(%)
Świrski [30]	23.33	19.33	77.17	86.17	80.83	55.00
I	10.27	5.48	5.11	70.95	85.41	3.01
II	0.00	4.16	26.34	34.26	64.55	1.98
III	3.34	1.71	6.81	39.44	63.57	11.63
IV	4.03	4.44	34.54	81.58	84.03	3.31
V	7.73	14.66	77.85	77.28	84.07	1.83
VI	10.30	19.14	19.34	53.18	77.00	7.18
VII	1.70	2.41	39.35	46.91	59.08	0.29
VIII	7.62	9.52	41.90	56.83	66.19	9.21
ExCuSe [5] IX	5.93	13.88	24.09	74.60	86.47	6.36
X	22.98	51.07	29.88	79.76	82.38	64.05
XI	14.66	27.79	20.31	56.49	75.27	17.40
XII	26.72	64.50	71.37	79.20	79.39	74.24
XIII	33.40	46.64	61.51	70.26	73.52	32.18
XIV	0.21	22.81	53.30	57.57	84.22	76.76
XV	26.17	7.71	60.88	52.34	57.58	44.35
XVI	26.02	8.93	17.86	49.49	68.11	45.66
XVII	0.00	0.75	70.90	77.99	89.93	85.07
XVIII	1.32	1.92	12.39	22.24	49.76	2.16
XIX	2.68	5.25	9.03	26.45	32.50	5.29
XX	2.18	3.73	17.93	52.37	67.65	0.94
ElSe [6] XXI	0.54	2.41	8.09	43.54	42.61	0.01
XXII	0.64	5.91	1.98	27.93	48.66	0.21
XXIII	55.43	8.03	96.54	93.86	93.55	15.88
XXIV	0.94	1.88	44.43	45.21	52.97	1.04
XXV	56.86	39.79	84.48	63.53	87.95	65.58
XXVI	48.68	19.70	41.58	29.90	69.87	24.72
XXVII	27.55	6.75	31.43	34.83	57.50	21.82
XXVIII	7.70	9.27	16.87	25.38	37.42	14.53
XXIX	6.75	0.00	8.38	19.08	22.95	13.15
XXX	11.10	13.30	63.48	53.44	84.10	38.72
XXXI	43.55	7.65	66.17	66.48	73.60	68.43
XXXII	42.17	34.32	77.68	75.32	81.00	64.22
XXXIII	35.65	30.90	56.40	60.42	61.97	45.20
XXXIV	10.42	3.65	71.23	59.00	72.65	41.62
LPW [32] XXXV	31.07	18.10	31.58	49.52	71.48	9.45
XXXVI	54.92	24.10	71.82	72.58	89.73	49.58
XXXVII	14.75	16.52	27.03	45.04	51.51	16.68
XXXVIII	30.25	23.50	76.07	58.60	70.50	57.52
XXXIX	27.17	8.15	37.80	44.83	53.95	43.78
XXXX	23.24	17.24	74.11	72.73	82.13	82.01
XXXXI	20.90	2.42	68.88	42.10	72.97	47.52
XXXXII	50.67	33.48	61.18	66.25	78.57	48.73
XXXXIII	11.97	3.45	24.87	21.88	54.05	2.60
XXXXIV	19.83	16.58	41.40	11.72	83.35	0.98
XXXXV	30.43	25.63	55.90	47.45	88.92	28.18
XXXXVI	41.60	11.85	6.48	31.23	70.00	0.83

Table 2 Performance comparison of SET, Starburst, Świrski, ExCuSe and ElSe in terms of detection rate up to an error of five pixels. The best performance on each data set is shown in bold.

Figure 11 shows evaluation results for the most challenging data sets. Data set XIX in Figure 11(a) is characterized by scattered reflections, which lead to edges on the pupil but not at its boundary. Since most of the state-of-the-art approaches are based on the edge

filtering, they are very likely to fail in detecting the pupil boundary. In consequence, the detection rates achieved here are quite poor.

Data set XXI (Figure 11(b)) poses challenges related to poor illumination conditions, leading thus to an iris with low intensity values.

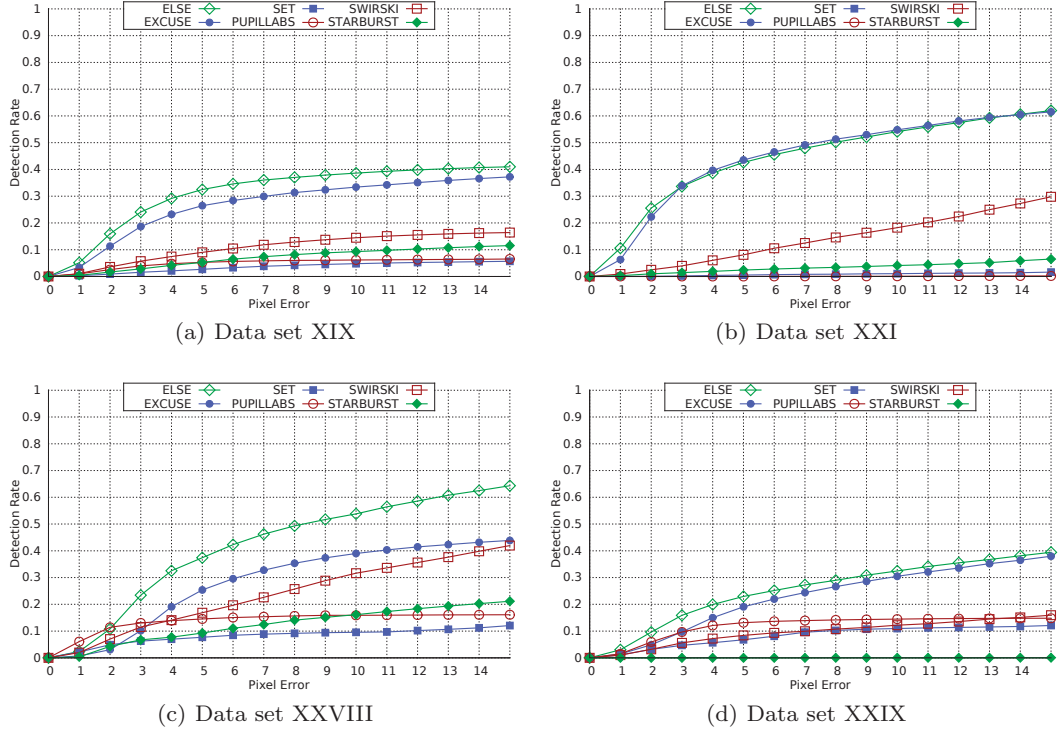


Fig. 11 Limitations of the state-of-the-art algorithms.

This makes it very difficult to separate the pupil from the iris (e.g., in the Canny edge detection the responses are discarded because they are too low). Additionally, this data set contains reflections, which have a negative impact on the edge filter response. While the algorithms ElSe and ExCuSe achieve detection rates of approximately 45%, the remaining approaches can detect the pupil center in only 10% of the eye images. Figure 12 presents examples of successfully found pupils in eye images from Data set XXI. The top row shows two input images, the middle row presents the filtered edges, and the bottom row shows the pupils detected by the ElSe algorithms. Among the evaluated algorithms, only ElSe [6] and ExCuSe [5] find the pupil in these eye images. The remaining algorithms fail due to the low contrast in the pupil area. More specifically, SET [8] fails, since in the thresholding step large parts of the iris are extracted and identified as pupil area. Świrski et. al [30] fails in the coarse positioning step,

while Starburst fails while selecting the correct edge candidates that represent the pupil border.

The eye images contained in Data set XXVIII (Figure 11(c)) are recorded from a highly off-axial camera position. In addition, poor illumination makes it difficult to separate the pupil from the dark regions at the eyelid areas. Both conditions lead to overall poor detection rates. Figure 13 shows failure cases of ElSe [6] on eye images from Data sets XIX and XXVIII. To demonstrate the challenges associated with automated pupil detection in these images we have chosen ElSe because it was the best performing algorithm. The left column presents the input images to the algorithms, the second column shows the filtered edges, the third column are the blob responses, and the last column the results. The first two rows contain images from Data set XIX and show the high impact of scattered reflections (first row) and induced curved edges by reflections (second row). In

the second, row the third image is not present since ElSe [6] did not use blob detection. The last row shows an input image from Data set XXVIII, where the wrong blob response is due to eyelashes. Since the image is recorded highly off-axis, the pupil is only marginally available.

The last challenging Data set XXIX (Figure 11(d)) is also characterized by highly off-axial images. In addition, the frame of the subjects' glasses covers the pupil and most of the images are heavily blurred. This leads to unsatisfactory responses from the Canny edge detector. In consequence, the detection rates are very poor, e.g. ElSe (the best performing algorithm) can detect the pupil in only 25% of the eye im-

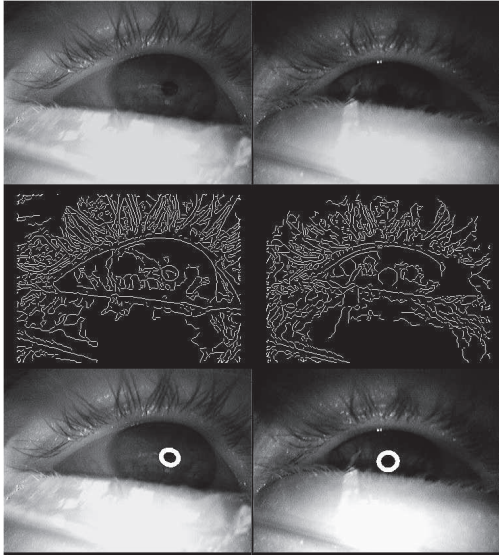


Fig. 12 Successfully found pupils by the ElSe algorithm in eye images from Data set XXI.

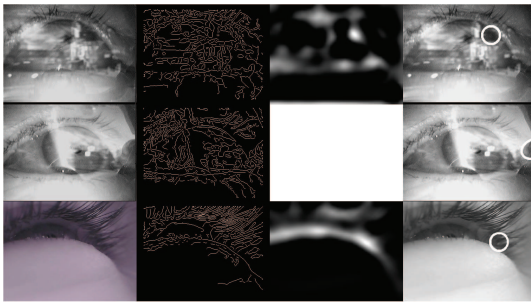


Fig. 13 Failure cases of ElSe on eye images from Data sets XIX and XXVIII.

ages. Figure 14 shows failure cases of ElSe [6] on eye images from Data sets XXI and XXIX. The left column presents the input images, the second column shows the filtered edges, the third column are the blob responses, and the last column the results obtained with the ElSe algorithm. The blob response of ElSe [6] in the top row is distracted by the high surface difference of the light shadow at the lower eyelid and the bright skin below it. For Data set XXIX, the main pupil recognition problems arise due to the bright eyeglasses frame, which distracts the blob response. In the above failure cases, further improvements in automated pupil detection could come from explicitly considering additional eye-related features such as eyelids and eye corners.

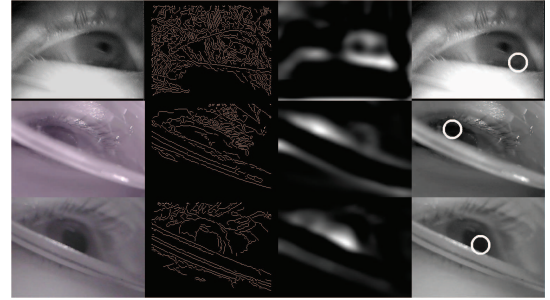


Fig. 14 Failure cases of ElSe on eye images from Data sets XXI and XXIX.

5 Conclusions

We presented a review of state-of-the art pupil detection algorithms for application in outdoor settings. The focus was primarily on the robustness of these algorithms with respect to frequently and rapidly changing illumination conditions, off-axial camera position, and other sources of noise. Six state-of-the-art approaches were evaluated on over 200,000 ground-truth annotated images collected with different eye-tracking devices in a range of different everyday settings. Our extensive evaluation shows that despite good average performance of these algorithms on these challenging data sets, there

are still problems in obtaining robust pupil centers in case of reflections or poor illumination conditions.

6 Acknowledgements

This work was funded, in part, by the Cluster of Excellence on Multimodal Computing and Interaction (MMCI) at Saarland University as well as a JST CREST research grant.

References

1. Braunagel C, Kasneci E, Stolzmann W, Rosenstiel W (2015) Driver-activity recognition in the context of conditionally autonomous driving. In: 2015 IEEE 18th International Conference on Intelligent Transportation Systems, pp 1652–1657, DOI 10.1109/ITSC.2015.268
2. Bulling A, Ward JA, Gellersen H, Tröster G (2011) Eye Movement Analysis for Activity Recognition Using Electrooculography. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33(4):741–753, DOI 10.1109/TPAMI.2010.86
3. Bulling A, Weichel C, Gellersen H (2013) Eyecontext: Recognition of high-level contextual cues from human visual behaviour. In: Proc. of the 31st SIGCHI International Conference on Human Factors in Computing Systems (CHI), pp 305–308, DOI 10.1145/2470654.2470697
4. Douglas DH, Peucker TK (1973) Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization* 10(2):112–122
5. Fuhl W, Kübler T, Sippel K, Rosenstiel W, Kasneci E (2015) ExCuSe: Robust Pupil Detection in Real-World Scenarios. In: *Computer Analysis of Images and Patterns*, Springer, pp 39–51
6. Fuhl W, Santini TC, Kübler T, Kasneci E (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*, ACM, New York, NY, USA, ETRA '16, pp 123–130
7. Goni S, Echeto J, Villanueva A, Cabeza R (2004) Robust algorithm for pupil-glint vector detection in a video-oculography eye-tracking system. In: *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, IEEE
8. Javadi AH, Hakimi Z, Barati M, Walsh V, Tcheang L (2015) Set: a pupil detection method using sinusoidal approximation. *Frontiers in neuroengineering* 8
9. Jian M, Lam KM (2015) Simultaneous hallucination and recognition of low-resolution faces based on singular value decomposition. *Circuits and Systems for Video Technology, IEEE Transactions on* 25(11):1761–1772
10. Jian M, Lam KM, Dong J (2013) A novel face-hallucination scheme based on singular value decomposition. *Pattern Recognition* 46(11):3091–3102
11. Jian M, Lam KM, Dong J (2014) Facial-feature detection and localization based on a hierarchical scheme. *Information Sciences* 262:1–14
12. Kasneci E (2013) Towards the automated recognition of assistance need for drivers with impaired visual field. PhD thesis, University of Tübingen, Wilhelmstr. 32, 72074 Tübingen, URL <http://tobias-lib.uni-tuebingen.de/volltexte/2013/7033>
13. Kasneci E, Sippel K, Aehling K, Heister M, Rosenstiel W, Schiefer U, Papageorgiou E (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):e87,470
14. Kasneci E, Sippel K, Heister M, Aehling K, Rosenstiel W, Schiefer U, Papageorgiou E (2014) Homonymous visual field loss and its impact on visual exploration: A super-market study. *TVST* 3(6)

15. Kasneci E, Kasneci G, Kübler TC, Rosenstiel W (2015) *Artificial Neural Networks: Methods and Applications in Bio-/Neuroinformatics*, Springer International Publishing, chap Online Recognition of Fixations, Saccades, and Smooth Pursuits for Automated Analysis of Traffic Hazard Perception, pp 411–434. DOI 10.1007/978-3-319-09903-3_20
16. Kassner M, Patera W, Bulling A (2014) Pupil: an open source platform for pervasive eye tracking and mobile gaze-based interaction. In: Adj. Proc. of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp), pp 1151–1160, DOI 10.1145/2638728.2641695
17. Keil A, Albuquerque G, Berger K, Magnor MA (2010) Real-time gaze tracking with a consumer-grade video camera
18. Li D, Winfield D, Parkhurst DJ (2005) Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches. In: Computer Vision and Pattern Recognition-Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on, IEEE, pp 79–79
19. Lin L, Pan L, Wei L, Yu L (2010) A robust and accurate detection of pupil images. In: Biomedical Engineering and Informatics (BMEI), 2010 3rd International Conference on, IEEE, vol 1, pp 70–74
20. Liu X, Xu F, Fujimura K (2002) Real-time eye detection and tracking for driver observation under various light conditions. In: Intelligent Vehicle Symposium, 2002. IEEE, IEEE, vol 2, pp 344–351
21. Long X, Tonguz OK, Kiderman A (2007) A high speed eye tracking system with robust pupil center estimation algorithm. In: Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE, IEEE
22. Majaranta P, Bulling A (2014) *Eye Tracking and Eye-Based Human-Computer Interaction*, Springer Publishing London, pp 39–65. *Advances in Physiological Computing*, DOI 10.1007/978-1-4471-6392-3_3
23. Mohammed GJ, Hong BR, Jarjes AA (2012) Accurate pupil features extraction based on new projection function. *Computing and Informatics* 29(4):663–680
24. Pérez A, Cordoba M, Garcia A, Méndez R, Munoz M, Pedraza JL, Sanchez F (2003) A precise eye-gaze detection and tracking system
25. Schnipke SK, Todd MW (2000) Trials and tribulations of using an eye-tracking system. In: CHI'00 extended abstracts on Human factors in computing systems, ACM
26. Sippel K, Kasneci E, Aehling K, Heister M, Rosenstiel W, Schiefer U, Papageorgiou E (2014) Binocular Glaucomatous Visual Field Loss and Its Impact on Visual Exploration - A Supermarket Study. *PLoS ONE* 9(8):e106089, DOI 10.1371/journal.pone.0106089, URL <http://dx.doi.org/10.1371/journal.pone.0106089>
27. Stellmach S, Dachsel R (2012) Look & touch: gaze-supported target acquisition. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, pp 2981–2990
28. Sugano Y, Bulling A (2015) Self-calibrating head-mounted eye trackers using egocentric visual saliency. In: Proc. of the 28th ACM Symposium on User Interface Software and Technology (UIST), pp 363–372, DOI 10.1145/2807442.2807445
29. Suzuki S, et al (1985) Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing* 30(1):32–46
30. Świrski L, Bulling A, Dodgson N (2012) Robust real-time pupil tracking in highly off-axis images. In: Proceedings of the Symposium on Eye Tracking Research & Applications (ETRA), ACM, pp 173–176, DOI 10.1145/2168556.2168585
31. Tafaj E, Kübler T, Kasneci G, Rosenstiel W, Bogdan M (2013) Online classification of eye tracking data for automated analysis of traffic hazard perception. In: *Artificial Neural Networks and Machine Learning*

- ICANN 2013, Springer Berlin Heidelberg, vol 8131, pp 442–450
32. Tonsen M, Zhang X, Sugano Y, Bulling A (2016) Labelled pupils in the wild: A dataset for studying pupil detection in unconstrained environments. In: Proc. of the ACM International Symposium on Eye Tracking Research & Applications (ETRA), pp 139–142, DOI 10.1145/2857491.2857520
 33. Trösterer S, Meschtscherjakov A, Wilfinger D, Tscheligi M (2014) Eye tracking in the car: Challenges in a dual-task scenario on a test track. In: Proceedings of the 6th AutomotiveUI, ACM
 34. Turner J, Bulling A, Alexander J, Gellersen H (2014) Cross-device gaze-supported point-to-point content transfer. In: Proceedings of the ACM International Symposium on Eye Tracking Research & Applications (ETRA), pp 19–26, DOI 10.1145/2578153.2578155
 35. Valenti R, Gevers T (2012) Accurate eye center location through invariant isocentric patterns. *Transactions on pattern analysis and machine intelligence* 34(9):1785–1798
 36. Viola P, Jones M (2001) Rapid object detection using a boosted cascade of simple features. In: *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, IEEE, vol 1, pp I–511
 37. Wood E, Bulling A (2014) Eyetab: Model-based gaze estimation on unmodified tablet computers. In: Proc. of the 8th Symposium on Eye Tracking Research & Applications (ETRA), pp 207–210, DOI 10.1145/2578153.2578185
 38. Zhu D, Moore ST, Raphan T (1999) Robust pupil center detection using a curvature algorithm. *Computer methods and programs in biomedicine* 59(3):145–157