

Link to data:

<https://atreyus.informatik.uni-tuebingen.de/seafiler/d/8e2ab8c3fdd444e1a135/?p=%2FDataset%20and%20Algos%20remote&mode=list>

Evaluation of State-of-the-Art Pupil Detection Algorithms on Remote Eye Images

Wolfgang Fuhl

Perception Engineering
University of Tübingen, Germany
wolfgang.fuhl@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

Thiago Santini

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

Abstract

Eye movements are a powerful source of information as well as the most intuitive form of interaction. Although eye-tracking technology is still in its infancy, it offers the greatest potential for novel communication solutions and applications. Whereas head-mounted eye-trackers are widely used in research, several applications require most unintrusive eye tracking, ideally realized by means of a single, low-cost camera placed away from the subject. However, such remote devices usually provide low resolution images and pose several challenges to gaze position estimation. The key challenge in such a scenario is the robust detection of the pupil center in the recorded image. We evaluated eight state-of-the-art algorithms for pupil detection on three manually labeled data sets recorded in remote tracking scenarios. Among the evaluated algorithms, EISe [6] proved to be the best performing approach on overall 3202 images from remote eye tracking, which include changing illumination, occlusion, head movements, and off-axial camera position. In addition, we contribute a new data set with 445 annotated images, recorded in a fixed setup with a low cost camera capable of using natural and infrared light.

Author Keywords

Pupil detection, remote eye tracking, data set, algorithm comparison

Paste the appropriate copyright statement here. ACM now supports three different copyright statements:

- ACM copyright: ACM holds the copyright on the work. This is the historical approach.
- License: The author(s) retain copyright, but ACM receives an exclusive publication license.
- Open Access: The author(s) wish to pay for the work to be open access. The additional fee must be paid to ACM.

This text field is large enough to hold the appropriate release statement assuming it is single spaced in a sans-serif 7 point font.

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

ACM Classification Keywords

I.4.0 [General]: Image processing software; I.4.8 [Scene Analysis]: Object recognition

Introduction

Video-based eye tracking is available as head-mounted and remote technology. In remote eye tracking, one or more cameras are mounted in the scene and aligned to the subject's head. The gaze can be estimated from the recorded images after several processing steps based on a correct extraction of the eye and, in particular, of the pupil center. Most image-based pupil detection algorithms are developed for head-mounted eye-tracking technology or remote technology, where the cameras zoom to the subject's eyes [5, 6, 9, 12, 14]. Many use cases, such as in the context of driver observation (e.g., [3, 13]) or gaze-based assistance systems (e.g., [11]), require accurate and non-intrusive systems. In such use cases, cameras are placed at some distance from the subject, mostly recording the upper body of the subject.

Applying the above pupil detection algorithms to remote tracking requires some prior steps to extract the eye region from the recorded image. This could lead to impairments in pupil detection due to inaccuracy in the eye region detection. In addition, the characteristics of the extracted eye regions are quite different compared to head-mounted setups. Usually, they provide a lower resolution as well as a reduced color depth. Besides the image quality, remote systems also need to deal with inadequate camera perspectives or eyes not visible in the image due to extreme head angles.

In this work, we evaluate the performance of pupil detection algorithms on image data from remote tracking. Eight approaches from the state of the art in head-mounted and remote eye-tracking video processing are evaluated on two

existing data sets, namely BioID [10] and GI4E [17], and on a new data set introduced with this work. By applying these algorithms on automatically detected and manually labeled eye regions, we investigate and discuss the influence of inaccuracy related to eye region detection.

This evaluation, including algorithms and data, is available for download at:

<http://www.ti.uni-tuebingen.de/Pupil-detection.1827.0.html>

Eye-image processing algorithms

In a recent publication, Fuhl et al. [7] evaluated state-of-the-art algorithms for pupil detection in eye images from head-mounted eye tracking. For our evaluation on low resolution images, we employ the algorithms evaluated there, namely Swirski et al. [14], Starburst [12], ExCuSe [5], SET [9], and the best performing approach EISE [6]. In addition, we evaluate methods that were explicitly design to process image data from remote tracking, such as the approach by George and Routray [8], Droege and Paulus [4], and by Timm and Barth [15]. For each of these algorithms, we choose the best parameter setting based on the BioID. In the following, we will briefly describe each of the algorithms.

Swirski

Swirski et al. [14] estimate the pupil contour in three stages. In the first stage, Haar-like features are applied to obtain the pupil region in the input image, i.e., to reduce the search space. The second stage consists of a segmentation of the pupil region by using a k-means clustering of the histogram followed by a Canny edge detection. Finally, RANSAC is applied to estimate the pupil contour, for which the edge gradient direction of the inliers are considered [14].

ExCuSe

ExCuSe [5] is based on edge detection and ellipse fitting, combining two algorithms for pupil detection. Depending

on the histogram analysis one of them will be applied. The first algorithm estimates a coarse pupil center. A Canny filter provides an edge image, which is then refined by morphological operations and finally filtered by analyzing the curvature of the edges. An ellipse approaching the pupil contour is finally fit to the best edge. The second algorithm detects a coarse pupil position by an angular integral projection function and refines this position iteratively until convergence. On each iteration rays are spread out from the last determined pupil center in every direction in the Canny filtered input image. If a ray hits an edge, the cutting point is used to fit an ellipse, refines thus the pupil center position [5].

SET

SET operates based on the assumption that the pupil shape is circular. Therefore, it processes a binary image by first thresholding the image and, afterwards, employs segmentation to group related pixels. Segments containing less than a certain number of pixels are left out. For each remaining segment, the border pixels are calculated and used as input for an ellipse fitting. The center of the ellipse which is closest to a circle is taken to represent the pupil center [9].

EISe

Similar to ExCuSe, EISe provides two algorithms to estimate the pupil center. The first algorithm uses a Canny filtered image and morphological operations to detect pupil-related edges. As ExCuSe, it selects the best edge by various heuristics like the shape and enclosing intensity. This edge is used to perform an ellipse fitting, which yields the pupil center and contour. In case the first algorithm is not successful, an advanced blob detection is applied to find the pupil center [6].

Starburst

Starburst is an iterative method to estimate the pupil contour on an input image. It starts at any random start point in the image and sends rays in all directions. If a ray exceeds a minimal edge threshold, the related point in the image is marked. In the second step, all marked points sending a burst of rays back into the direction of the origin and also marks the first point the ray exceed the edge threshold. All the points marked in this way are used to find the best fitting ellipse. There can be several marked points which are not related to the pupil contour. Therefore, Starburst uses RANSAC for ellipse fitting to remove outliers. The center point of the fitted ellipse is used as the new start point for the next iteration [12].

Droege and Paulus

In their approach, Droege and Paulus [4] consider the fact that when the eye region is extracted from images recorded by a remote camera, not only the image is of low resolution but may contain glints covering parts of the pupil or iris. Therefore, the authors use the direction and length of the pixel gradients as a stable feature. More specifically, the pixel gradients are first filtered to remove irrelevant pixel gradients or outliers. For estimating the pupil center the intersection of all remaining gradient vectors is then calculated by applying an M-Estimator¹ [4].

Timm and Barth

Similar to Droege and Paulus, Timm and Barth use the direction of pixel gradients as a feature to determine the pupil center [15]. The basic idea of this approach is that the direction from the pupil center point to any pupil or iris contour point should be the same as the gradient direction at the pupil or iris contour (except for the sign). The pupil cen-

¹Our implementation of the algorithm from Droege and Paulus solves the equation system using least square instead of a M-Estimator

ter is determined as following: for each pixel in the input image, the algorithm runs over the whole gradient image and accumulates the square product of the normalized displacement vector of the pupil center candidate to the gradient pixel and the normalized pixel gradient vector. To take into account that the pupil is usually dark, Timm and Barth use an additional factor that multiplies the accumulator by the inverted intensity of the pupil center candidate. Finally, the center candidate with the highest accumulator is taken as the pupil center [15].

George and Routray

The approach by George and Routray [8] starts with a coarse positioning by using the orientation anulus convolution filter from [1]. The filter initialization was slightly modified to adjust the weights of vertical edge gradients stronger. Furthermore, the authors employ a Schaar kernel for calculating a gradient image, followed by an inverted circular mean filter to weight dark regions. All local maxima in the result are then selected, and the relationship between the response, standard deviation, and the mean value is calculated. The local maximum is selected as coarse position. Afterward, all gradients in the radius range are thresholded based on their magnitude and angle to the coarse position point vector. The collected gradients are median filtered, and an RANSAC ellipse fit is applied. In our implementation of this algorithm, we apply a least squares ellipse fit to all instances instead of RANSAC-based ellipse fitting since it showed better results empirically.

Data sets

We evaluated the previously described algorithms on three data sets, namely BioID [10], GI4E [17], and our own dataset.

BioID

The BioID data set consists of 1521 grayscale images (384×286 pixel) showing faces of 23 subjects recorded at different office environments with varying illumination. Challenges associated with this dataset are different camera distances to the subjects, blink and small head movements. Figure 1 shows example images out of the BioID data set. The data set was collected for evaluation of face detection algorithms. Hence, BioID provides only information about the eye centers, but there are several further labels (pupil center and various facial landmarks, such as eye corner or nose tip) available for this data set provided by the FGnet project.



Figure 1: Example images from the BioID data set.

GI4E

The GI4E data set provides 1236 RGB images (800×600 pixel) of 103 subjects recorded by a standard webcam. Each image shows a frontal face view. For each subject, 12 sequential frames in which the subject gazes at different points on a screen are provided. Some of the subjects wear glasses. Figure 2 shows four example images randomly taken from the data set. For each image, the eye corners, and the iris center is labeled [17].



Figure 2: Example images from the GI4E data set.

New data set

With this work, we provide a new data set consisting of 445 manually-labeled images of two subjects. This data was recorded in an office environment by means of a FOSCAM FL9826P surveillance camera and contains distinct head and eye movements as well as blinks and reflections. The data set includes infrared and grayscale converted RGB images of 1280×960 pixel resolution. For each frame, we labeled the iris and pupil contour and the bounding box around each eyes. Figure 3 shows some example images from this data set.



Figure 3: Example images from our new data set. The left two images are grayscale converted RGB images. The remaining ones were infrared recorded.

Evaluation procedure

Detecting eye regions in remote data is a non-trivial task and usually produces eye regions with varying shape, size, and position. In contrast, labeled eye regions are usually aligned to the eye corners or other landmarks like eyebrows by definition of the labeling procedure. To examine the influence of these differences, our evaluation is conducted for two scenarios: in the first case, labeled eye regions were extracted and used as input to a pupil detection algorithm; in the second scenario the pre trained openCV [2] Haar Cascade [18] combined with a KLT-feature tracking [16] is applied for eye region extraction. We removed all misclassified eye regions for a fair comparison of algorithms. Eye regions in remote data are usually of low resolution compared to head-mounted setups. Since the above algorithms Starburst, Swirsky, EISe, ExCuSe, and SET were designed for high resolution eye images, we scaled up the extracted

	BioID		GI4E		New	
	Man. labeled	HC detected	Man. labeled	HC detected	Man. labeled	HC detected
Minimum	20x40	12x18	20.5x20.9	26x39	22x30	35.1x49.7
Maximum	20x40	35x52	26.7x44.9	42x62	47x99	79x118
Mean	20x40	21.2x31.8	22x31.2	30.1x45.2	24.5x60.8	49.8x74.5
Median	20x40	21x32	21.5x30.7	30x45	22x61	49x74

Table 1: The eye region resolutions in pixel for all data sets including the detected eye boxes.

eye regions from remote images using a nearest neighbor interpolation. For each algorithm, we explored the best parameter setting based on the achieved performance on the BioID dataset. This setting was then used for further evaluation.

We report the detection rate relative to an error tolerance given by the euclidean distance between ground truth and detected pupil center. Additionally the normed error, dividing the euclidean error by the diagonal length of the eye region, is shown too.

In table 1 the minimum, maximum, mean and median values are shown. The minimum is the resolution of the minimal diagonal. Maximum and median resolutions are also calculated based on the diagonal.

Results

We evaluated the algorithms both on the detected eye regions by means of a Haar-Cascade and on annotated eye regions. As shown in Figure 4, the Haar Cascade achieved 2774 successful detections on the BioID data set, 2437 on the GI4E and 534 on our new one. In contrast do the data sets BioID and GI4E, our data contains a large number of head rotations and occluded eyes, which reduces the number of successfully detected eye region considerably.

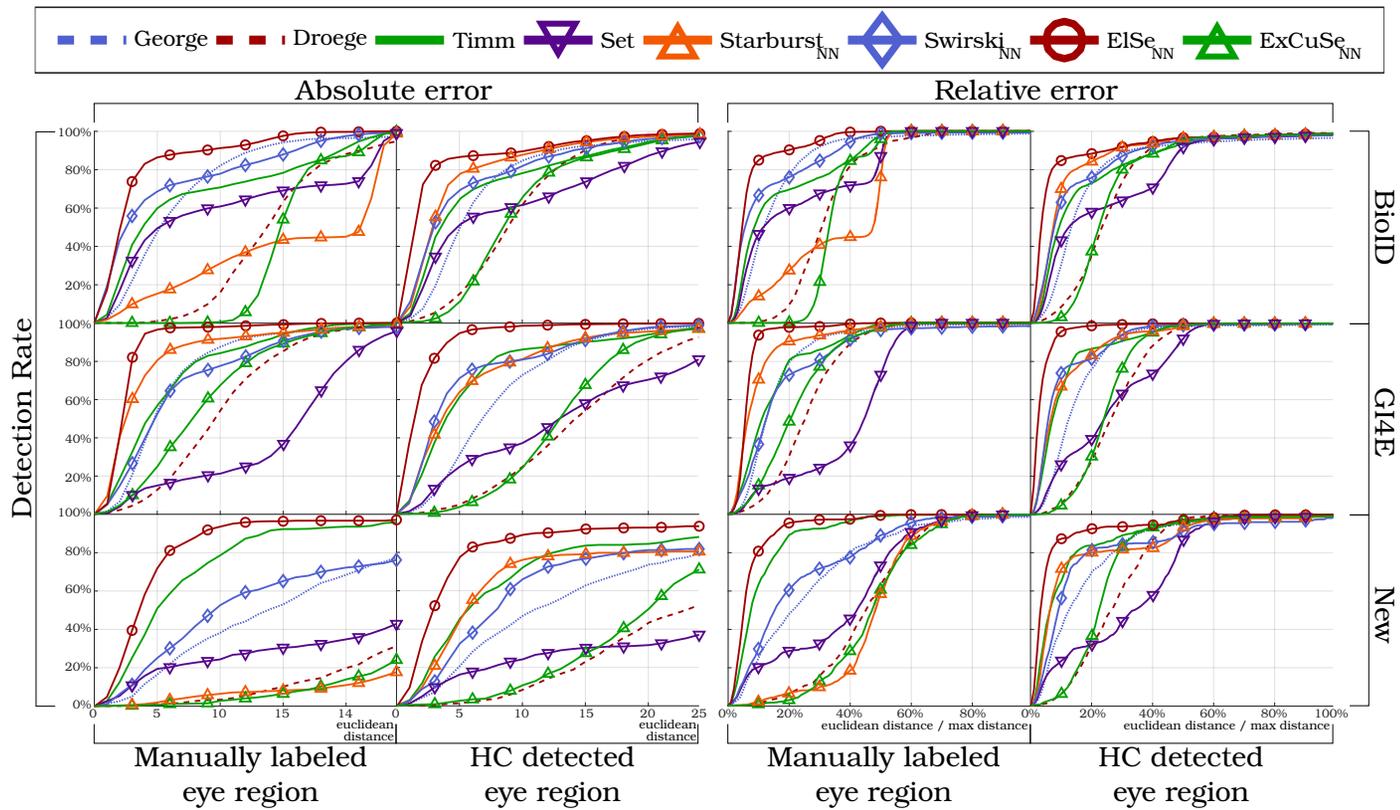


Figure 5: The left sided six plots (title absolute error) show the euclidean distance in pixels whereas the right sided six plots (title relative error) show it normed with the eye box diagonal. In the top box the membership of the color to the algorithm is defined. The first and third column show the results for the labeled eye boxes and the second and fourth column the results for the detected eye boxes. Each data set is show separately as indicated on the left.

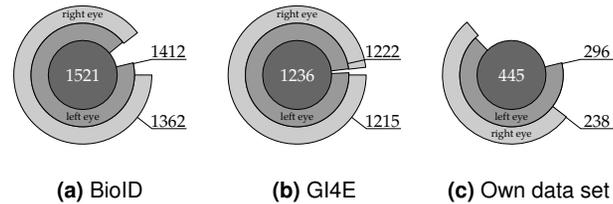


Figure 4: Proportion of correct detected eye regions by means of Haar Cascades.

Figure 5 shows the results of the evaluation. The plots on the left side provide the detection rate depending on the euclidean distance between the predicted pupil center and ground truth. The plots on the right side of the image are based on a relative error measure, i.e., the euclidean distance is normalized by the maximum error (diagonal length of the eye region). Thus, effects that arise due to different sizes of the processed eye regions are compensated. Note that each row of plots in Figure 5 corresponds to the results on one data set.

Our results show that algorithms designed for processing remote images do not achieve better results than algorithms designed for head-mounted tracking. Among the evaluated approaches, the algorithm EISe [6] and the approach by Timm and Barth [15] show a stable detection rate on all data sets, which suggests that these algorithms are robust against various sources of noise, such as illumination or off-axial camera position.

Table 2 provides the detection rate of the evaluated algorithms at a relative error of 20%. The best performing algorithms on the BioID data set are EISe and Starburst. In terms of the relative error, EISe achieves detection rates of 90.7% at a relative error of 20%. Similar results are

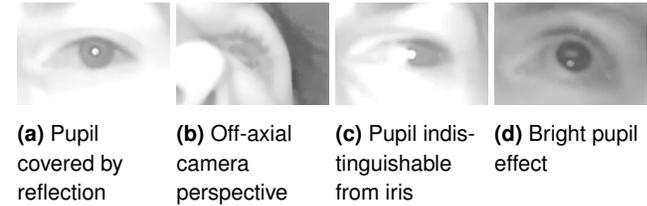


Figure 6: Challenges posed by our new data set.

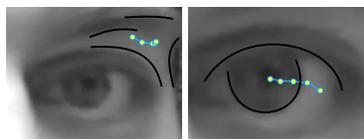
achieved on the GI4E data set, where EISe has very high detection rates (close to 100%) on the manually labeled bounding boxes. Lower detection rates are achieved by all algorithms on our own data set. On the manually labeled bounding boxes, EISe and the algorithm by Timm and Barth show very high detection rates of above 90%. Lower detection rates are achieved when the eye regions were detected by the Haar-Cascade. At a relative error of 20%, the algorithm by Timm and Barth, EISe, and Swirski reach similar detection rates of approximately 80%. This data set is especially challenging due off-axial eye images caused by head movements as well as reflections on the pupil and other effects caused by the infrared lighting as shown in Figure 6.

Although specifically designed to process images in the context of remote eye tracking, the algorithms by George and Routray [8] and Droege and Paulus [4] show lower detection rates than the best performing algorithms. Similar performance is also achieved by the algorithms for head-mounted eye tracking ExCuSe [5] and SET [9]. Especially in the case of Starburst, we observe an algorithmic performance that is highly dependent on the size of the processed eye region (manually labeled vs. detected by a Haar Cascade). For Starburst we selected the center of the image as start point. This position is improved in each iter-

	BioID		GI4E		New	
	Man. labeled	HC detected	Man. labeled	HC detected	Man. labeled	HC detected
George	0.772	0.884	0.794	0.760	0.479	0.703
Droege	0.096	0.739	0.304	0.283	0.066	0.316
Timm	0.696	0.823	0.806	0.867	0.891	0.838
SET	0.599	0.638	0.189	0.395	0.288	0.323
Starburst	0.273	0.915	0.904	0.835	0.066	0.802
Swirski	0.759	0.799	0.729	0.815	0.605	0.814
EISe	0.907	0.939	0.983	0.898	0.927	0.933
ExCuSe	0.011	0.796	0.484	0.311	0.037	0.065

Table 2: Detection rate of all evaluated algorithms with a relative error tolerance of 20%

ation by selecting surrounding edges and model fitting. In general this works well in case the start point is inside the eye. The larger the eye region is, the more likely it is that Starburst selects an inadequate start point and converges to a local minima that is unlikely to contain the pupil. Figure 7 exemplarily shows how starburst fails in a large eye region (a) and succeeds (b) in a smaller eye region.



(a) Large eye region with inadequate start point

(b) Small eye region with adequate start point

Figure 7: The blue line indicates the star point and the following estimated pupil center by the algorithm Starburst. The black lines indicate ray reflecting edges.

Most algorithms designed for head-mounted setups, pursuing different approaches for detecting pupil edges. Eye images extracted from a remote camera usually have a

poor image quality compared to head-mounted setups as shown on Figure 6 which makes edge detection difficult. EISe shows the best results but on closer examination we found that only the second algorithm (blob detection) were successful. The evaluated algorithms designed for remote data uses pixel gradients instead of edges as more stable feature in bad image quality than edges.

Conclusion

Remote eye tracking deals with the challenge of pupil detection in low-resolution images and off-axial camera perspectives. Only few pupil detection algorithms address these points. In this work we evaluated eight state-of-the-art algorithms regarding their capability to find the pupil center on remote data. Our evaluation reveals that algorithms designed for head-mounted setups, especially EISe [6] can not only be applied to pupil detection in remote data but even outperform them. In addition, we introduced a new annotated data set for pupil detection on remote data. This data set provides 445 images with distinct head movements, blinks, and varying illumination collected by a low-cost surveillance camera using RGB and infrared. In our future work, we will consider the pupil shape in the error measure, since in many use-case, a valid pupil shape could be helpful for 3D reconstruction.

All newly implementations of algorithms and the datasets are available at:

<http://www.ti.uni-tuebingen.de/Pupil-detection.1827.0.html>

REFERENCES

1. T. J. Atherton and D. J. Kerbyson. 1999. Size invariant circle detection. *Image and Vision computing* 17, 11 (1999), 795–803.
2. G. Bradski. 2000. OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000).

3. 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. DOI: <http://dx.doi.org/10.1109/ITSC.2015.268>
4. D. Droege and D. Paulus. 2010. Pupil center detection in low resolution images. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*. ACM, 169–172.
5. 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.
6. W. Fuhl, T. C. Santini, T. Kübler, and E. Kasneci. 2016a. 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.
7. W. Fuhl, M. Tonsen, A. Bulling, and E. Kasneci. 2016b. Pupil detection for head-mounted eye tracking in the wild: an evaluation of the state of the art. In *Machine Vision and Applications*. Springer.
8. A. George and A. Routray. 2016. Fast and Accurate Algorithm for Eye Localization for Gaze Tracking in Low Resolution Images. *arXiv preprint arXiv:1605.05272* (2016).
9. A.-H. Javadi, Z. Hakimi, M. Barati, V. Walsh, and L. Tcheang. 2015. SET: a pupil detection method using sinusoidal approximation. *Frontiers in neuroengineering* 8 (2015).
10. O. Jesorsky, K. J Kirchberg, and Robert W. F. 2001. Robust face detection using the hausdorff distance. In *Audio-and video-based biometric person authentication*. Springer, 90–95.
11. E. Kasneci. 2013. *Towards the Automated Recognition of Assistance Need for Drivers with Impaired Visual Field*. Ph.D. Dissertation. University of Tübingen, Wilhelmstr. 32, 72074 Tübingen. <http://tobias-lib.uni-tuebingen.de/volltexte/2013/7033>
12. D. Li, D. Winfield, and D. J. Parkhurst. 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, 79–79.
13. X. Liu, F. Xu, and K. Fujimura. 2002. Real-time eye detection and tracking for driver observation under various light conditions. In *Intelligent Vehicle Symposium, 2002. IEEE*, Vol. 2. IEEE, 344–351.
14. L. Świrski, A. Bulling, and N. Dodgson. 2012. Robust real-time pupil tracking in highly off-axis images. In *Proceedings of the Symposium on Eye Tracking Research & Applications (ETRA)*. ACM, 173–176. DOI: <http://dx.doi.org/10.1145/2168556.2168585>
15. F. Timm and E. Barth. 2011. Accurate Eye Centre Localisation by Means of Gradients. *VISAPP* 11 (2011), 125–130.
16. C. Tomasi and T. Kanade. 1991. *Detection and tracking of point features*. School of Computer Science, Carnegie Mellon Univ. Pittsburgh.

17. A. Villanueva, V. Ponz, L. Sesma-Sanchez, M. Ariz, S. Porta, and R. Cabeza. 2013. Hybrid method based on topography for robust detection of iris center and eye corners. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 9, 4 (2013), 25.
18. P. Viola and M. Jones. 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*, Vol. 1. IEEE, 1–511.