Computer Vision and Image Understanding
journal homepage: www.elsevier.com

PuRe: Robust pupil detection for real-time pervasive eye tracking

Thiago Santini[a,2,**], Wolfgang Fuhl[a,2], Enkelejda Kasneci[a]

[a] *University of Tübingen, Sand 14, Tübingen – 72076, Germany*

ABSTRACT

Real-time, accurate, and robust pupil detection is an essential prerequisite to enable pervasive eye--tracking and its applications – e.g., gaze-based human computer interaction, health monitoring, foveated rendering, and advanced driver assistance. However, automated pupil detection has proved to be an intricate task in real-world scenarios due to a large mixture of challenges such as quickly changing illumination and occlusions. In this paper, we introduce the *Pupil Reconstructor* (*PuRe*), a method for pupil detection in pervasive scenarios based on a novel edge segment selection and conditional segment combination schemes; the method also includes a confidence measure for the detected pupil. The proposed method was evaluated on over 316,000 images acquired with four distinct head–mounted eye tracking devices. Results show a pupil detection rate improvement of over 10 percentage points w.r.t. state-of-the-art algorithms in the two most challenging data sets (6.46 for all data sets), further pushing the envelope for pupil detection. Moreover, we advance the evaluation protocol of pupil detection algorithms by also considering eye images in which pupils are not present and contributing a new data set of mostly closed eyes images. In this aspect, *PuRe* improved *precision* and *specificity* w.r.t. state-of-the-art algorithms by 25.05 and 10.94 percentage points, respectively, demonstrating the meaningfulness of *PuRe*'s confidence measure. *PuRe* operates in real-time for modern eye trackers (at 120 *fps*) and is fully integrated into *EyeRecToo* – an open-source state-of-the-art software for pervasive head-mounted eye tracking. The proposed method and data set are available at `www.ti.uni-tuebingen.de/perception`.

1. Introduction

Head-mounted video-based eye trackers are becoming increasingly more accessible and prevalent. For instance, such eye trackers are now available as low-cost devices (e.g., Pupil Labs (2017)) or integrated into wearables such as Google Glasses, Microsoft Hololens, and the Oculus Rift (Raffle and Wang, 2015; Microsoft, 2017; Oculus, 2017). As a consequence, eye trackers are no longer constrained to their origins as research instruments but are developing into fully fledged pervasive devices. Therefore, guaranteeing that these devices are able to seamlessly operate in *out-of-the-lab* scenarios is not only pertinent to the research of human perception, but also to enable further applications such as pervasive gaze-based human-computer interaction (Bulling and Gellersen, 2010), health monitoring (Vidal et al., 2012), foveated rendering (Guenter et al., 2012), and conditionally automated driving (Braunagel et al., 2017).

Pupil detection is the fundamental layer in the eye-tracking stack since most other layers rely on the signal generated by this layer – e.g., for gaze estimation (Morimoto and Mimica, 2005), and automatic identification of eye movements (Santini et al., 2016). Thus, errors in the pupil detection layer propagate to other layers, systematically degrading eye-tracking performance. Unfortunately, robust real-time pupil detection in natural environments has remained an elusive challenge. This elusiveness is evidenced by several reports of difficulties and low pupil detection rates in natural environments such as driving (Schmidt et al., 2017; Wood et al., 2017; Kübler et al., 2015; Trösterer et al., 2014; Kasneci, 2013; Chu et al., 2010; Liu et al., 2002), shopping (Kasneci et al., 2014), walking (Sugano and Bulling, 2015; Foulsham et al., 2011), in an operating room (Tien et al., 2015), and during human-robot interac-

[**] Corresponding author:
e-mail: `thiago.santini@uni-tuebingen.de` (Wolfgang Fuhl)

[2] Authors contributed equally and should be considered co-first authors.

tion (Aronson et al., 2018). These difficulties in pupil detection stems from multiple factors; for instance, reflections (Fig. 1a), occlusions (Fig. 1b), complex illuminations (Fig. 1c), and physiological irregularities (Fig. 1d) (Fuhl et al., 2016d; Hansen and Hammoud, 2007; Hansen and Pece, 2005; Zhu and Ji, 2005).

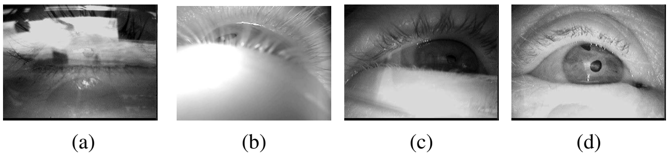


Fig. 1: Representative images of pupil detection challenges in real-world scenarios: (a) reflections, (b) occlusions, (c) complex illuminations, and (d) physiological irregularities.

In this paper, we introduce the *Pupil Reconstructor* (*PuRe*), a method for pupil detection in pervasive scenarios based on a novel edge segment selection and conditional segment combination schemes; the proposed method also includes a meaningful confidence measure for the detected pupil. Previous work in the field of pupil detection is presented in Section 2, and the proposed method is described in Section 3. Previous work usually focuses on evaluating pupil detection algorithms based solely on the detection rate. Similarly, we contrast the proposed method to previous work in Section 4.1. In addition, we introduce novel metrics to evaluate these algorithms in terms of incorrect pupil detection rates (Section 4.2) as well as dynamic signal properties (Section 4.3). Run time considerations are discussed in Section 4.4, and Section 6 presents final remarks and future work.

2. Related Work

While there is a plethora of previous work for pupil detection, most methods are not suitable for *out-of-the-lab* scenarios. For an extensive appraisal of state-of-the-art pupil detection methods, we refer the reader to the works by Fuhl et al. (2016d) and Tonsen et al. (2016) for head-mounted eye trackers as well as Fuhl et al. (2016a) for remote eye trackers. In this work, we focus solely on methods that have been shown to be robust enough for deployment in pervasive scenarios, namely *ElSe* (Fuhl et al., 2016c), *ExCuSe* (Fuhl et al., 2015), and *Świrski* (Świrski et al., 2012).

ElSe consists of two approaches. First, a Canny edge detector is applied, and the resulting edges are filtered through morphological operations[3]. Afterwards, ellipses are fit to the remaining edges, edges are removed based on empirically defined heuristics, and one ellipse is selected as pupil based on its roundness and enclosed intensity value. If this method fails to produce a pupil, a second approach that combines a mean and a center surround filter to find a coarse pupil estimate is employed; an area around this coarse estimate is then thresholded with an adaptive parameter, and the center of mass of pixels below the threshold is returned as pupil center estimate (Fuhl et al., 2016c).

ExCuSe first analyzes the input images w.r.t. reflections based on peaks in the intensity histogram. If the image is determined to be reflection free, the image is thresholded with an adaptive parameter, and a coarse pupil position is estimated through an angular integral projection function Mohammed et al. (2012); this position is then refined based on surrounding intensity values. If a reflection is detected, a Canny edge detector is applied, and the resulting edges are filtered with morphological operations; ellipses are fit to the remaining edges, and the pupil is then selected as the ellipse with the darkest enclosed intensity (Fuhl et al., 2015).

Świrski starts with a coarse positioning using Haar-like features. The intensity histogram of an area around the coarse position is clustered using *k-means* clustering, followed by a modified *RANSAC*-based ellipse fit (Świrski et al., 2012).

From these algorithms, *ElSe* has shown a significantly better performance over multiple data sets (Fuhl et al., 2016d). Moreover, it is worth noticing that these algorithms employ multiple parameters that were empirically defined, albeit there is usually no need to tune these parameters.

It is worth dedicating part of this section to discuss machine-learning approaches in contrast to the algorithmic ones, particularly convolutional neural networks (CNN). Similarly to other computer vision problems, from a solely pupil detection stand point, deep CNNs will likely outperform human-crafted pupil detection approaches given enough training data – with incremental improvements appearing as more data becomes available and finer network tuning. Besides labeled data availability, which might be alleviated with developments of unsupervised learning methods, there are other impediments to the use of CNNs in pervasive scenarios since these scenarios typically require the use of embedded systems. For instance, computation time and power consumption. While these impediments might be lessened with specialized hardware – e.g., *cuDNN* (Chetlur et al., 2014), *Tensilica Vision DSP* (Efland et al., 2016), such hardware might not always be available or may incur prohibitive additional production costs. Finally, CNN-based approaches might be an interesting solution from an engineering point of view, but remain a *black box* from the scientific one. To date, we are aware of two previous works that employ CNNs for pupil detections: 1) *PupilNet* (Fuhl et al., 2016b), which aims at a computationally inexpensive solution in the absence of hardware support, and 2) *Vera-Olmos* (Vera-Olmos and Malpica, 2017), which consists of two very deep CNNs – a coarse estimation stage (with 35 convolution plus 7 max-pooling layers for encoding and 10 convolution plus 7 deconvolution layers for decoding), and a fine estimation stage (with 14 convolution plus 5 max-pooling layers for encoding and 7 convolution plus 5 deconvolution layers).

[3] Fuhl et al. (2016c) also describe an algorithmic approach to edge filtering producing similar results; however the morphological approach is preferred because it requires less computing power.

3. *PuRe*: The Pupil Reconstructor

Similarly to related work, the proposed method was designed for *near-infrared*[4] eye images acquired by head-mounted eye trackers. Our method only makes two uncomplicated assumptions to constrain the valid pupil dimension space without requiring empirically defined values: 1) the eye canthi lie within the image, and 2) the eye canthi cover at least two-thirds of the image diagonal. It is worth noticing that these are *soft* assumptions – i.e., the proposed method still operates satisfactorily if the assumptions are not significantly violated. Fig. 2 illustrates these concepts. Furthermore, these assumptions are in accordance to eye tracker placement typically suggested by eye tracker vendor's guidelines to capture the full range of eye movements.

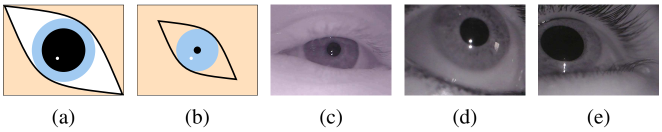


Fig. 2: *PuRe* assumptions visualized. (a) illustrates the maximal intercanthal distance, yielding the maximal pupil diameter (pd_{max}), whereas (b) illustrates the lower bound – i.e., minimal pupil diameter (pd_{min}). (c) shows realistic data that respects both assumptions. In contrast, the maximal intercanthal distance assumption is violated in (d) and (e). In the former, the pupil does not approach maximal dilation, and *PuRe* is still able to detect the pupil. In the latter, the pupil is significantly dilated, and the resulting diameter exceeds pd_{max}; *PuRe* does not detect such pupils.

PuRe works *purely* based on edges, selecting curved edge segments that are likely to be significant parts of the pupil outline. These selected segments are then conditionally combined to construct further candidates that may represent a reconstructed pupil outline. An ellipse is fit to each candidate, and the candidate is evaluated based on its ellipse aspect ratio, the angular spread of its edges relative to the ellipse, and the ratio of ellipse outline points that support the hypothesis of it being a pupil. This evaluation yields a confidence measure for each candidate to be the pupil, and the candidate with the highest confidence measure is then selected as pupil. The remainder of this section describes the proposed method in detail.

3.1. Preprocessing

Prior to processing, if required, the input image is downscaled to the working size $S_w = (W_w \times H_w)$ through bilinear interpolation, where W_w and H_w are the working width and height, respectively. The original aspect ratio is respected during downscaling. Afterwards, the resulting image is linearly normalized using a *Min-Max* approach.

3.2. Edge Detection and Morphological Manipulation

PuRe's first step is to perform edge detection using a Canny edge operator (Canny, 1986). The resulting edge image is then manipulated with a morphological approach to thin and straighten edges as well as to break up orthogonal connections following the procedure described by Fuhl et al. (2016c). The result of this step is an image with unconnected and thinned edge segments, as illustrated in Fig. 3.

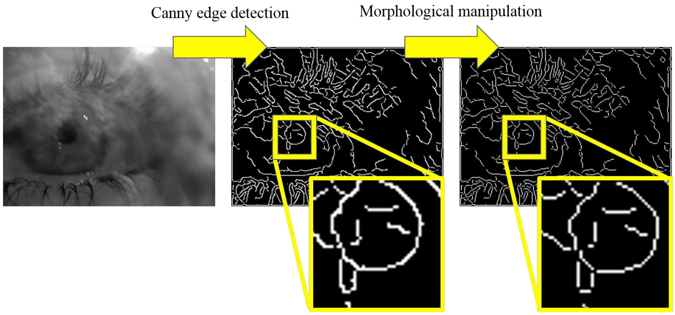


Fig. 3: Input image (left), resulting Canny edge detection (middle), and edges after morphological manipulation. Notice how the edges are thinned and orthogonal connections are broken.

3.3. Edge Segment Selection

Each edge segment is first approximated by a set of dominant points D following the *k-cosine* chain approximation method described by Teh and Chin (1989). This approximation reduces the computational requirements for our approach and typically results in a better ellipse fit in cases where a pupil segment has not been properly separated from surrounding edges. After approximation, multiple heuristics are applied to discard edge segments that are not likely to be part of the pupil outline:

1. Given the general conic equation $ax^2 + by^2 + cxy + dx + ey + f = 0$, at least five points are required to fit an ellipse in a least-squares sense. Therefore, we exclude segments in which D's cardinality is smaller than five. This heuristic discards plain shapes such as small segments and substantially straight lines.
2. Based on the assumptions highlighted in the beginning of this section, it is possible to establish the maximal and minimal distance between the lateral and medial eye canthus in pixels when frontally imaged as

   $$ec_{max} = \sqrt{{W_w}^2 + {H_w}^2} \quad \text{and} \quad ec_{min} = \frac{2}{3} * ec_{max}. \qquad (1)$$

   These estimates can then be used to infer rough values for the maximal (Fig. 2a) and minimal (Fig. 2b) pupil diameter bounds (pd_{max} and pd_{min}) based on the human physiology. We approximate the eye canthi distance through the palpebral fissure width as 27.6 mm (Kunjur et al., 2006); similarly, the maximal and minimal pupil diameter are approximated as 8 mm and 2 mm, respectively (Spector, 1990). Therefore,

   $$pd_{max} \approx 0.29 * ec_{max} \quad \text{and} \quad pd_{min} \approx 0.07 * ec_{min}. \qquad (2)$$

   Note that whereas maximal values hold independent of camera rotation and translation w.r.t. the eye, minimal values might not hold due to perspective projection distortions and corneal refractions. Nonetheless, pd_{min} already

[4] This is the standard image format for head-mounted eye trackers and can be compactly represented as a grayscale image.

represents a minute part (≈ 4.8%) of the image diagonal, and we opted to retain this lower bound – for reference, see Fig. 2b. For each candidate, we approximate the segment's diameter by the largest gap between two of its points. Candidates with a diameter outside of the range $[pd_{min}, pd_{max}]$ violate bounds and are thus discarded.

3. To estimate a segment's curvature, first the minimum rectangle containing D is calculated using the *rotating calipers* method (Toussaint, 1983). The curvature is then estimated based on the ratio between this rectangle's smallest and largest sides. The straighter the candidate is, the smaller the ratio. The cut-off threshold for this ratio is based on the ratio between the minor and major axes of an ellipsis with axes extremities inscribed in 45° of a circle, which evaluates to $R_{th} = (1 - cos(22.5°))/sin(22.5°) \approx 0.2$. This heuristic servers to discard relatively linear candidates.
4. At this stage, an ellipse E is fit to the points in D following the least-squares method described in Fitzgibbon and Fisher (1995). A segment is discarded if: I) E's center lies outside of the image boundaries, which violates *PuRe*'s assumptions, or II) the ratio between E's minor and major axes is smaller than R_{th}, which assumes that the camera pose relative to the eye can only distort the pupil round shape to a certain extend.
5. Seldom, the ellipse fitting procedure will not produce a proper fit. We identify and discard most such cases inexpensively if the mean point from D does not lie within the polygon defined by the extremities of E's axes.

As a result from this elimination process, the edge segment search space is significantly reduce, as illustrated by Fig. 4.

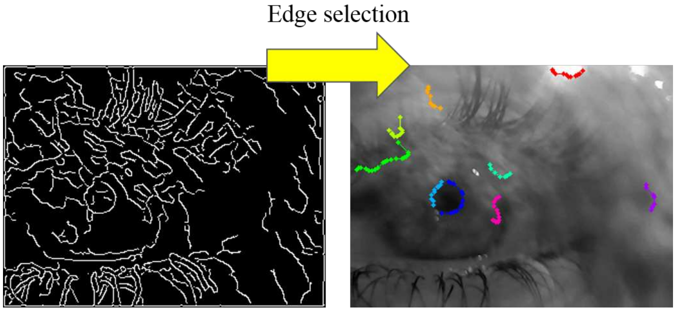


Fig. 4: Edges after morphological processing (left) and the resulting selected segments that are candidates for the pupil outline (right). Each segment is represented by its *k-cosine* chain approximation and illustrated with a distinct color.

3.4. Confidence Measure

For each remaining candidate, *PuRe* takes into account three distinct metrics to determine a confidence measure ψ that the candidate is a pupil:

Ellipse Aspect Ratio (ρ): measures the roundness of E. This metric favors rounder ellipses (that typically result due to the eye camera placement w.r.t. the eye) and is evaluated as the ratio between E's minor and major axis.

Angular Edge Spread (θ): measures the angular spread of the points in D relative to E, assuming that the better distributed the edges are, the more likely it is that the edges originated from a clearly defined elliptical shape (i.e., a pupil's shape). This metric is roughly approximated as the ratio of E centered quadrants that contain a point from D.

Ellipse Outline Contrast (γ): measures the ratio of the E's outline that supports the hypothesis of a darker region surrounded by a brighter region (i.e., a pupil's appearance). This metric is approximated by selecting E's outline points with a stride of ten degrees. For each point, the linear equation passing through the point and the E's center is calculated, which is used to define a line segment with length proportional to E's minor axis and centered at the outline point. If the mean intensity of the inner segment is lower than the mean intensity of the outer one, the point supports the pupil-appearance hypothesis[5].

If the candidate's ellipse outline is invalid – i.e., violates *PuRe*'s size assumptions or less than half of the outline contrast γ supports the candidate – the confidence metric is set to zero. Otherwise, the aforementioned metrics are averaged when determining the resulting confidence. In other words,

$$\psi = \begin{cases} 0 & \text{if the outline is invalid;} \\ \frac{\rho+\theta+\gamma}{3} & \text{otherwise.} \end{cases} \quad (3)$$

It is worth noticing that the range of all three metrics (and consequently ψ) is [0,1].

3.5. Conditional Segment Combination

The segments that remain as candidates are combined pairwise to generate additional candidates. This procedure attempts to reconstruct the pupil outline based on nearby segment pairs since the pupil outline is often broken up due to occlusions from, for example, reflections or eye lashes. Let D_1 and D_2 be the set of dominant points for two segments and S_1 and S_2 the set of points contained by the up-right squares bounding D_1 and D_2, respectively. The segments are combined if these bounding squares intersect but are not fully contained within one another – i.e., $S_1 \cap S_2 \neq \emptyset \neq S_1 \neq S_2$. For instance, see Fig. 5. The resulting merged segment is then validated according to Section 3.3, and its confidence measure evaluated according to Section 3.4. Since this procedure is likely to produce candidates with high aspect ratio ρ and angular spread θ values, the new candidate is only added to the candidate list if its outline contrast γ improves on the γ from the original segments. After conditional combination, the candidate with highest confidence ψ is selected as the initial pupil, as shown in Fig. 6.

Note that the inner intensities relative to other candidates do not contribute to the pupil selection. Thus, the iris might be selected since it exhibits properties similar to the pupil – e.g., roundness, inner-outer contrast, and size range. For this reason,

[5] If a bright pupil eye tracker is used, the inverse holds

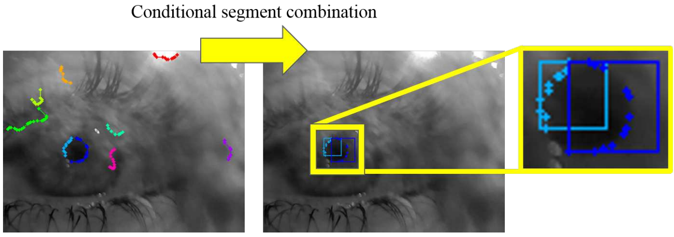


Fig. 5: Illustration of the conditional segment combination. The highlighted blue and cyan segments meet the intersection requirements and are combined to generate and additional pupil outline segment. The other pairs do not intersect and, therefore, generate no additional candidates.

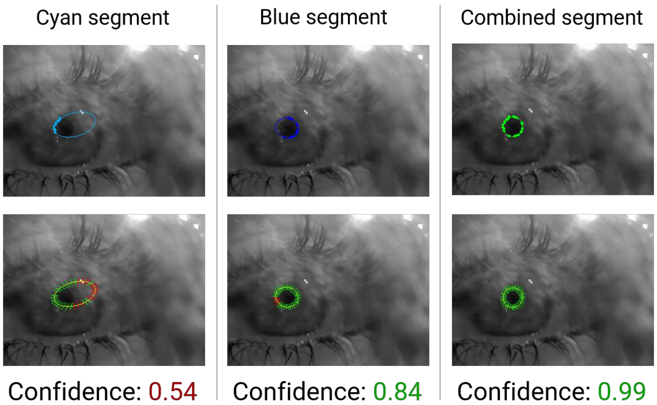


Fig. 6: Illustration for the confidence measure evaluation using the segments from Fig. 5. Segments with confidence smaller than 0.5 are omitted. The first row shows the segment points and resulting ellipse. Second row shows the lines contributing to the ellipse outline contrast (γ); green lines support the pupil-appearance hypothesis, whereas red lines do not. Notice how the cyan segment results in an incorrect outline estimation due to the ellipse fit even though the segment is part of the pupil outline. The blue segment results in an acceptable outline estimate even though the left side of the outline is slightly shifted. In contrast, the combined segment *reconstructs* the whole range of the pupil outline and yields a higher confidence (ψ), thus being selected as pupil estimate.

the inside of the initial pupil is searched for a roughly cocentered candidate with adequate size and strong inner-outer outline contrast. This is done by searching a circular area centered at the center of the initial pupil with radius equal to the initial semi-major axis – i.e., representing a circular iris. Candidates 1) lying inside this area, 2) with major axis smaller than the search radius, and 3) with at least three thirds of the outline contrast (γ) valid are collected. The collected candidate with highest confidence is then chosen as new the pupil estimate. If no candidate is collected in this procedure, the initial pupil remains as the pupil estimate. As output, *PuRe* returns not only a pupil center, but also its outline and a confidence metric.

4. Experimental Evaluation

As previously mentioned, we evaluate *PuRe* only against robust state-of-the-art pupil detection methods, namely *ElSe*[6], *ExCuSe*, and *Świrski*. All algorithms were evaluated using their open-source C++ implementations; default parameters were employed unless specified otherwise. For *ExCuSe*, the input images were downscaled to *240p* (i.e., 320 × 240 px) as there is evidence that this is a favorable input size detection-rate-wise (Tonsen et al., 2016). Similarly, the working size for *PuRe* (S_w, Section 3.1) was set to *240p* as well to keep run time compatible with state-of-the-art head-mounted eye trackers (see Section 4.4). *ElSe* provides an embedded downscaling and border cropping mechanism, effectively operating with a resolution of 346 × 260 px. Notice that whenever the input images are downscaled, the results must be upscaled to be compared with the ground truth. No preprocessing downscaling was performed for *Świrski* since evidence suggests it degrades performance for this method (Tonsen et al., 2016). Additionally, we juxtapose our results with the ones from *PupilNet* (Fuhl et al., 2016b) and *Vera-Olmos* (Vera-Olmos and Malpica, 2017) whenever possible.

In this work, we use the term **use case** to refer to each individual eye video. For instance, the *LPW* data set contains 22 subjects with three recordings per subject in distinct conditions (e.g., indoors, outdoors), resulting in 66 distinct *use cases*. Furthermore, we often compare *PuRe* with the **rival**, meaning the best performant from the other algorithms for the metric in question. For instance, for the aggregated detection rate, *ElSe* performs better than *ExCuSe* and *Świrski* and is, therefore, the *rival*.

4.1. Pupil Detection Rate

A pupil is considered detected if the algorithm's pupil center estimate lies within a radius of n pixels from the ground-truth pupil center. Similar to previous work, we focus on an error up to five pixels to account for small deviations in the ground-truth labeling process – e.g., human inaccuracy (Fuhl et al., 2015, 2016c; Tonsen et al., 2016; Vera-Olmos and Malpica, 2017). This error magnitude is illustrated in Fig. 7. For this evaluation, we employed five data sets totaling 266,786 realistic and challenging images acquired with three distinct head-mounted eye tracking devices, namely, the *Świrski* (Świrski et al., 2012), *ExCuSe* (Fuhl et al., 2015), *ElSe* (Fuhl et al., 2016c), *LPW* (Tonsen et al., 2016), and *PupilNet* (Fuhl et al., 2016b) data sets. In total, these data sets encompass 99 distinct *use cases*. It is worth noticing that we corrected[7] a disparity of one frame in the ground truth for five *use cases* of the *ElSe* data set and for the whole *PupilNet* data set, which increased the detection rate of all algorithms (by ≈ 3.5% on average).

Fig. 8 shows the cumulative detection rate per pixel error of the evaluated algorithms for the aggregated 266,786 images as well as the detection rate distribution per *use case* at five pixels. As can be seen, *PuRe* outperforms all algorithmic competitors for all pixel errors. In particular, *PuRe* achieved a detection rate of 72.02% at the five pixel error mark, further advancing the state-of-the-art detection rate by a significant margin of 6.46 percentage points when compared to the *rival*. Moreover, the

[6] With morphological split and validity threshold.

[7] All ground truth data employed in this work are available at `www.ti.uni-tuebingen.de/perception`

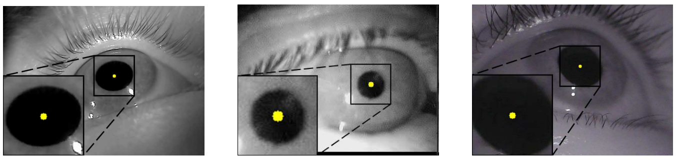


Fig. 7: Five pixels validity range (in yellow) around the ground-truth pupil center for the pupil estimate to be considered correct and, thus, the pupil detected. Reference range relative to the data from the *Świrski* (left), *ExCuSe/ElSe/PupilNet* (center), and *LPW* (right) data sets.

proposed method estimated the pupil center correctly 80% of the time for the majority of *use cases*, attesting for *PuRe*'s comprehensive applicability in realistic scenarios.

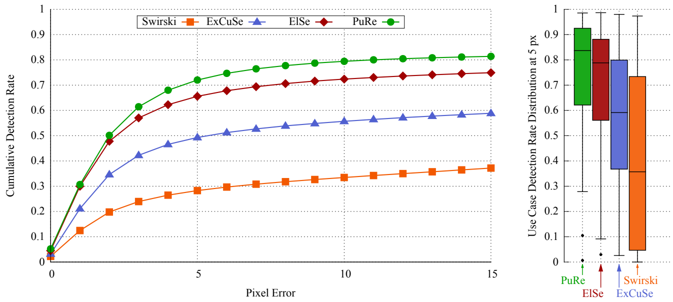


Fig. 8: On the left, the cumulative detection rate for the aggregated 266,786 images from all data sets. On the right, the distribution of the detection rate per *use case* as a *Tukey boxplot* (Frigge et al., 1989).

It is worth noticing that the aggregated detection rate does not account for differences in data set sizes. As a consequence, this metric is dominated by the *ExCuSe* and *LPW* datasets, which together represent 63.44% of the data. Since these two data sets are *not the most challenging ones*[8], the algorithms tend to perform better on them, and differences between the algorithms are less pronounced. Inspecting the detection rates per data set in Fig. 9 gives a better overview of the real differences between the algorithms and data sets, revealing that *PuRe* improves the detection rate by more than 10 percentage points w.r.t. the *rival* for the *most challenging data sets* (i.e., *ElSe* and *PupilNet*). To allow for a more fine-grained appreciation of the method's performance relative to the other algorithms, Fig. 10 presents *PuRe*'s detection rate at five pixels relative to the *rival* for each *use case*. In 71.72% of all *use cases*, *PuRe* outperformed all contenders. In particular, for the two most challenging data sets, *PuRe* surpassed the competition in 100% of the *use cases*. In contrast, the rivals noticeably outperformed *PuRe* in five *use cases*: `Swirski/p1-right`, `ExCuSe/data-set-II`, `LPW/4/12`, `LPW/9/17`, and `LPW/10/11`, from which representative frames are shown in Fig. 11. These five *use cases* also highlight some of *PuRe*'s imperfections. For instance, `Swirski/p1-right` and `ExCuSe/data-set-II` have weak and broken pupil edges due to inferior illumination and occlusions due to eye lashes/corneal reflections; *ElSe* compensates this lack of edges with its second step. `LPW/4/12` contains large pupils that violate *PuRe*'s assumptions; in fact, relaxing the maximum pupil size by only ten percent increases *PuRe*'s detection rate from 44.2% to 65.5% (or +6.55% w.r.t. the *rival*). `LPW/9/17` often has parts of the pupil outline occluded by eye lashes and reflections, whereas `LPW/10/11` contains pupils in extremely off-axial positions combined with occlusions caused by reflections. However, visually inspecting the latter two *use cases*, we did not find any particular reason for *Świrski* to outperform the other algorithms.

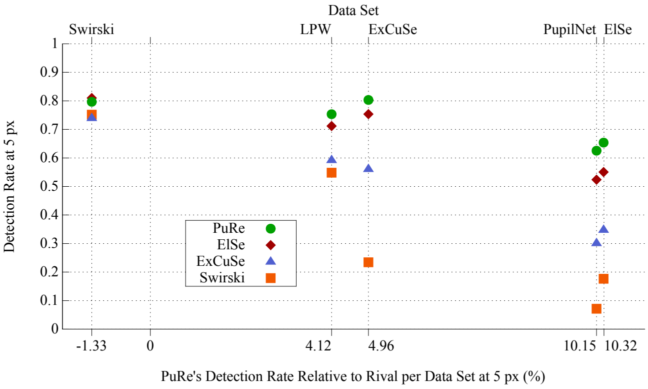


Fig. 9: Detection rate per data set plotted against *PuRe*'s performance relative to the *rival*. The lower the points, the harder the data set; the further right the points, the larger *PuRe*'s performance w.r.t. the *rival* is. Notice that as the data sets become more difficult (i.e., the detection rate decreases for all algorithms), the gap between *PuRe* and the other algorithms increases.

Regarding CNN-based approaches[9]: 1) For *PupilNet*, Fuhl et al. (2016b) report a detection rate of 65.88% at the five pixel error range when trained in half of the data from the *ExCuSe* and *PupilNet* data sets and evaluated on the remaining data. In contrast, *PuRe* reached 71.11% on all images from these data sets – i.e., +5.23%. 2) For *Vera-Olmos*, Vera-Olmos and Malpica (2017) report an unweighted[10] detection rate of 82.17% at the five pixel error range averaged over a *leave-one-out* cross validation in the *ExCuSe* and *ElSe* data sets. In contrast, *PuRe* reached 76.71% on all images from these data sets – i.e., −5.46%. Nevertheless, these results indicate that *PuRe* is able to compete with state-of-the-art CNN-based approaches while requiring only a small fraction of CNN computational requirements. In fact, *PuRe* outperformed *Vera-Olmos* for 37.5% of use cases. Furthermore, it is worth noticing that the training data is relatively similar to the evaluation data (same eye tracker, similar conditions and positioning) in both cases, which might bias the results in favor of the CNN approaches.

4.2. Beyond Pupil Detection Rate: Improving Precision, and Specificity Through the Confidence Measure

One aspect that is often overlooked when developing pupil detection algorithms is the rate of incorrect pupil estimates

[8] As evidenced by higher detection rates for all algorithms in Fig. 9

[9] These approaches used the uncorrected *ElSe* and *PupilNet* data sets, which might slightly affect the detection rate.

[10] Averaged over the *use cases*.

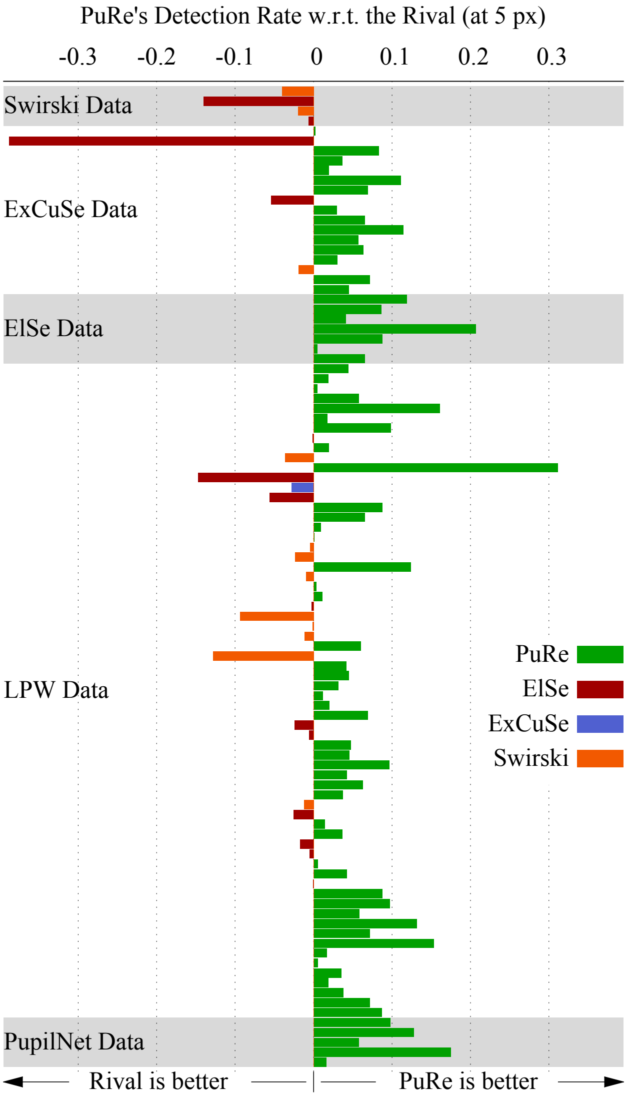


Fig. 10: *PuRe*'s performance relative to the rival for each *use case*. *PuRe* is the best algorithm in 71.72% of cases, *ElSe* in 14.14%, *Świrski* in 12.12%, and *ExCuSe* in 1.01%.

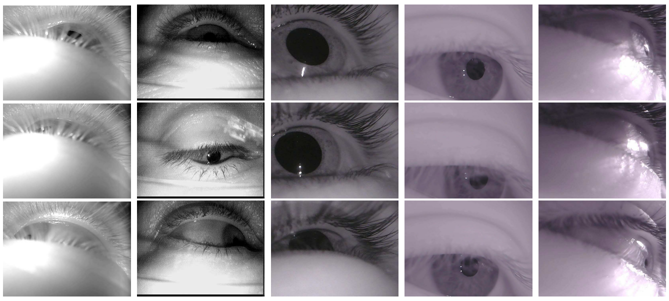


Fig. 11: Representative frames for *use cases* in which the *rival* outperforms *PuRe*. Each column contains frames from one *use case*. From left to right: `Swirski/p1-right`, `ExCuSe/data-set-II`, `LPW/4/12`, `LPW/9/17`, and `LPW/10/11`.

returned by the algorithm. For instance, the aforementioned CNN-based approaches *always* return a pupil estimate, regardless of one actually existing in the image. Intuitively, one can relax pupil appearance constraints in order to increase the detection rate, leading to an increase in the amount of incorrect pupils returned. However, these incorrect pupil estimates later appear as noise and can significantly degrade gaze-estimation calibration (Santini et al., 2017b), automatic eye movement detection (Pedrotti et al., 2011), glanced-area ratio estimations (Vrzakova and Bednarik, 2012), eye model construction (Świrski and Dodgson, 2013), or even lead to wrong medical diagnosis (Jansen et al., 2009). Therefore, it is imperative to also analyse algorithms in terms of incorrect detected pupils. The pupil detection task can be formulated as a classification problem – similar to the approach by Bashir and Porikli (2006) for frame-based tracking metrics – such that:

True Positive (*TP*) represents cases in which the algorithm and ground truth agree on the presence of a pupil. We further specialize this class into Correct True Positive (*CTP*) and Incorrect True Positive (*ITP*) following the detection definition from Section 4.1.

False Positive (*FP*) represents cases in which the algorithm finds a pupil although no pupil is annotated in the ground truth.

True Negative (*TN*) represents cases in which the algorithm and ground truth agree on the absence of a pupil.

False Negative (*FN*) represents cases in which the algorithm fails to find the pupil annotated in the ground truth.

Note that this is not a proper binary classification problem, and the *relevant* class is given only by *CTP*. Therefore, we redefine *sensitivity* and *precision* in terms of this class as

$$sensitivity = \frac{CTP}{TP + FN} \tag{4}$$

and

$$precision = \frac{CTP}{TP + FP} \tag{5}$$

respectively, such that *sensitivity* reflects the (correct) pupil detection rate and *precision* the rate of pupils that the algorithm found that are correct. Thus, these metrics allows us to evaluate 1) the trade-off between detection of correct and incorrect pupils, and 2) the meaningfulness of *PuRe*'s confidence measure.

Unfortunately, the eye image corpus employed to evaluate pupil detection rates (in Section 4.1) do not include negative samples – i.e., eye images in which a pupil is not visible, such as during a blink. Therefore, the capability of the algorithm to identify frames without a pupil as such cannot be evaluated since *specificity* ($\frac{TN}{TN+FP}$) remains undefined without negative samples. To evaluate this aspect of the algorithms, we have recorded a new data set (henceforth referred to as *Closed-Eyes*) containing in its majority (99.49%) negative samples. This data set consists of 83 *use cases* and contains 49,790 images with a resolution of 384 × 288 px. These images were collected from eleven subjects using a *Dikablis Professional* eye tracker (Ergoneers, 2017) with varying illumination conditions and camera positions. A larger appearance variation was achieved by asking the subjects to perform certain eye movement patterns[11] while their palpebrae remained shut in two conditions: 1) with the palpebrae softly shut, and 2) with the palpebrae strongly shut as to create additional skin folds. In ≈ 56% of *use cases*, participants wore glasses. Challenges in the images include reflections, black eyewear frames, prominent eye lashes, makeup, and skin folds, all of which can generate edge responses that the algorithms might identify as parts of the pupil outline. Fig. 12 shows representative images from the data set. This new data set with manually annotated ground truth (including pupil center whenever visible) is openly available at: `www.ti.uni-tuebingen.de/perception`

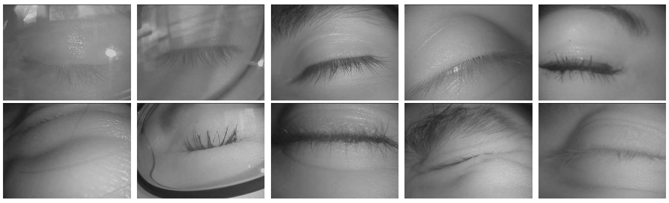


Fig. 12: Samples from the *Closed-Eyes* data set. First row shows samples from softly shut palpebrae, and the second one shows samples from strongly shut palpebrae.

We evaluated the four aforementioned algorithms using all images from the data sets from Section 4.1 and the *Closed-Eyes* data set, totaling 316,576 images. We assessed *PuRe*'s confidence measure using a threshold within [0:0.99] with strides of 0.01 units. A pupil estimate was considered correct only if its confidence measure was above the threshold. Similarly, *ElSe* offers a *validity threshold* (default=10) to diminish incorrect pupil rates, which we evaluated within the range [0:110] with strides of 10 units. *ExCuSe* and *Świrski* do not offer any incorrect pupil prevention mechanisms and, therefore, result only in a single evaluation point. The results from this evaluation are presented in Fig. 13. As can be seen in this figure, *PuRe* dominates over the other algorithms, and *PuRe*'s confidence metric is remarkably meaningful, allowing to significantly reduce incorrect pupil detections while preserving the correct pupil detection rate and increasing identification of frames without pupils. In fact, when compared to threshold 0, the threshold that maximizes the F_2 score (0.66) increased *precision* and *specificity* by 20.78% and 89.47%, respectively, whereas *sensitivity* was decreased by a negligible 0.49%. In contrast, *ElSe* exhibited negligible ($< 1\%$) changes for *sensitivity* and *precision* when varying the threshold from 0 to 10, with a small gain of 2.69% in *specificity*; subsequent increases in the threshold increase *specificity* at the cost of significantly deteriorating *ElSe*'s performance for the other two metrics. Compared to the *rival* for each metric, $PuRe_{th=0.66}$ improved *sensitivity*, *precision*, and *specificity* by 5.96, 25.05, and 10.94 percentage points, respectively.

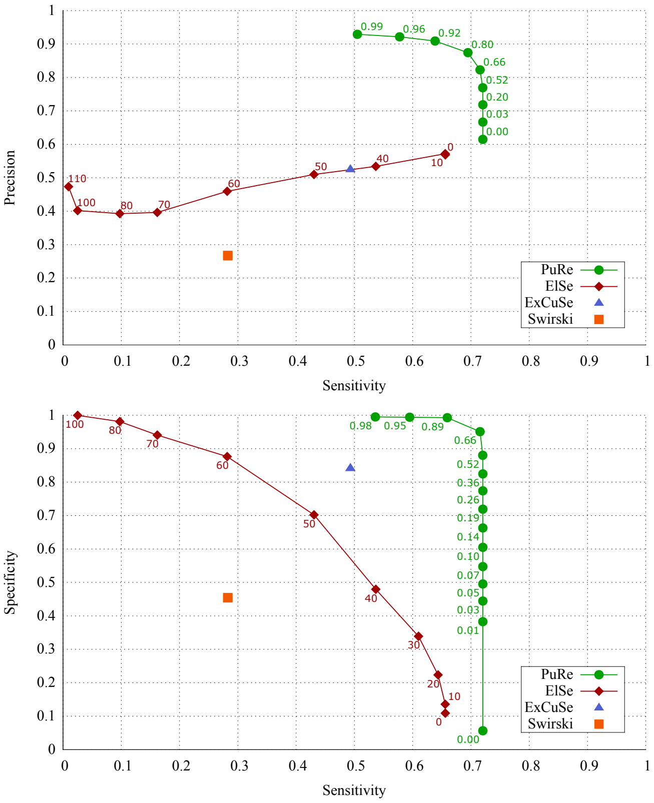


Fig. 13: Trade-off between *sensitivity* and *precision* (top) as well as *sensitivity* and *specificity* (bottom) for different pupil validation thresholds for *PuRe* and *ElSe*. Algorithms were evaluated over all images from the *Świrski*, *ExCuSe*, *ElSe*, *LPW*, *PupilNet*, and *Closed-Eyes* data sets. For the sake of visibility, points are only plotted when there's a significant (> 0.05) change in one of the metrics. The Z_2 score is maximized at thresholds 0.66 (for *PuRe*) and 10 (for *ElSe*).

4.3. Pupil Signal Quality

From the point of view of the image processing layer in the eye-tracking stack, the (correct/incorrect) detection rates stand as a meaningful metric to measure the quality of pupil detection algorithms. However, the remaining layers (e.g., gaze estimation, eye movement identification) often see the output of this layer as a discrete *pupil signal* (as a single-object tracking-by-detection), which these detection rates do not fully describe. For example, consider two pupil detection algorithms: A_1, which detects the pupil correctly every two frames, and A_2, which detects the pupil correctly only through the first half of the data. Based solely on the pupil detection rate (50% in both cases), these algorithms are identical. Nonetheless, the former algorithm enables noisy[12] eye tracking throughout the whole data, whereas the latter enables noiseless eye tracking during only the first half of the data. Which algorithm is preferable is then application dependent, but a method to assess these properties is required nonetheless.

[11] Although the eye is hidden underneath the palpebrae, eye globe movement results in changes in the folds and light reflections in the skin.

[12] Note that the values are not necessarily missing but might be incorrect pupil detections; thus interpolation/smoothing might actually degrade the pupil signal even further.

Recent analyses of widely-used object tracking performance metrics have shown that most existing metrics are strongly correlated and propose the use of only two weakly-correlated metrics to measure tracker performance: *accuracy* and *robustness* (Čehovin et al., 2016; Kristan et al., 2016). Whereas in those works *accuracy* was measured by average region overlaps, for pupil detection data sets, only the pupil center is usually available. Thus, we employ the center-error-based *detection rate* as accuracy measure. As an indicator of *robustness*, Čehovin et al. (2016) proposes the *failure rate* considering the tracking from a reliability engineering point of view as a supervised system in which an operator reinitializes the tracker whenever it fails. For the pupil signal, our formulation differs slightly since there is no operator reinitialization. Instead, we evaluate the *robustness* as the *reliability*

$$r = e^{-\lambda t}, \tag{6}$$

where $\lambda = \frac{1}{MTBF}$ is the failure rate estimated through the Mean Time Between Failures (*MTBF*) not accounting for *repair time* – i.e., periods of no/incorrect pupil detection are considered as latent faults. In this manner, the *reliability* is a measure of the likelihood of the algorithm correctly detecting the pupil for t successive frames. Furthermore, by measuring the Mean Time To Repair (*MTTR*) – i.e., the mean duration of periods in which the correct pupil signal is not available – we can achieve a similar metric in terms of the likelihood for the algorithm to *not* detect the pupil correctly for t successive frames. Henceforth, we will define this metric as the *insufficiency* (i), evaluated as

$$i = e^{-\kappa t}, \tag{7}$$

where $\kappa = \frac{1}{MTTR}$. The smaller an algorithm's *insufficiency*, the more *sufficient* it is. It is worth noticing, that r and i are not true probabilities since the events they measure are not likely to be independent nor uniformly distributed. Consequently, these metrics only offer a qualitative and relative measure between algorithms. Thus, we simplify their evaluation by fixing $t = 1$. As an illustration, let us return to our initial example considering a sequence of L frames: A_1 yields $r_{A_1} = i_{A_1} = e^{-1/1}$, whereas A_2 yields $r_{A_2} = i_{A_2} = e^{-1/(0.5L)}$. Since $\forall L > 2 \implies r_{A_1} < r_{A_2} \wedge i_{A_1} < i_{A_2}$, we can conclude that A_2 is more reliable but less sufficient w.r.t. A_1 for sequences longer than two frames. A quantitative conclusion is, however, not possible. For the sake of understandability, we further define *sufficiency* (s) as the complement of *insufficiency* such that

$$s = 1 - i. \tag{8}$$

In this manner, higher values are better for all metrics in this section.

We evaluated the four aforementioned algorithms in terms of *reliability* and *sufficiency* using only the data sets from Section 4.1. The *Closed-Eyes* data set was excluded since it is not realistic from the temporal aspect – i.e., users are not likely to have their eyes closed for extended periods of time. Furhermore, it is worth noticing that each *use case* from the *ExCuSe*, *ElSe*, and *PupilNet* data sets consists of images sampled throughout a video based on the pupil detection failure of a commercial eye tracker; these *use cases* can be seen as videos with a low and inconstant sampling rate. Results aggregated for all images are shown in Fig. 14 and indicate *PuRe* as the most reliable and sufficient algorithm. Curiously, the second most reliable algorithm was *Świrski*, indicating that during use cases in which it was able to detect the pupil, it produced a more stable signal than *ElSe* and *ExCuSe* – although its detection rate is much lower relative to the other algorithms for challenging scenarios. This lower detection rate reflects on the *sufficiency*, in which *Świrski* is the worst performer; *ElSe* places second, followed by *ExCuSe*. Furthermore, Fig. 15 details these results per *use case*. In this scenario, *PuRe* was the most reliable algorithm in 66.67% of the *use cases*, followed by *Świrski* (23.23%), *ElSe* (7.07%), and *ExCuSe* (3.03%). These results demonstrate that *PuRe* is more reliable not only when taking into account all images but also for the majority of *use cases*. This higher reliability also reflects on *PuRe*'s longest period of consecutive correct pupil detections, which contained 859 frames (in `LPW/21/12`). In contrast, the longest sequence for the *rival* was only 578 frames (*ExCuSe*, also in `LPW/21/12`). *ElSe*'s longest period was of 386 frames in `LPW/10/8`, for which *PuRe* managed 411 frames. In terms of *sufficiency*, *ElSe* had a small lead with 41.41% of *use cases*, closely followed by *PuRe* (40.40%); *ExCuSe* and *Świrski* were far behind, winning 14.14% and 4.04% of use cases, respectively. The advantage of *ElSe* here is likely due to its second pupil detection step, which might return the correct pupil during periods of mostly incorrect detections, fragmenting these periods into smaller ones.

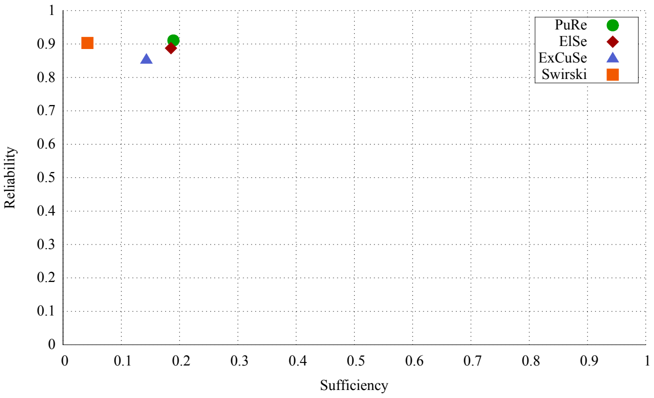


Fig. 14: *Reliability* and *sufficiency* for all algorithms based on the sequence of all aggregated images from the *Świrski*, *ExCuSe*, *ElSe*, *LPW*, and *PupilNet* data sets – higher is better.

4.4. Run Time

The run time of pupil detection algorithms is of particular importance for real-time usage – e.g., for human-computer interaction. In this section, we evaluate the temporal performance of the algorithms across all images from the *Świrski*, *ExCuSe*, *ElSe*, *LPW*, *PupilNet*, and *Closed-Eyes* data sets. Evaluation was performed on a Intel® Core^{TM} i5-4590 CPU @ 3.30GHz with 16GB RAM under Windows 8.1, which is similar to systems employed by eye tracker vendors. Results are shown in Fig. 16. All algorithms exhibited competitive performance in

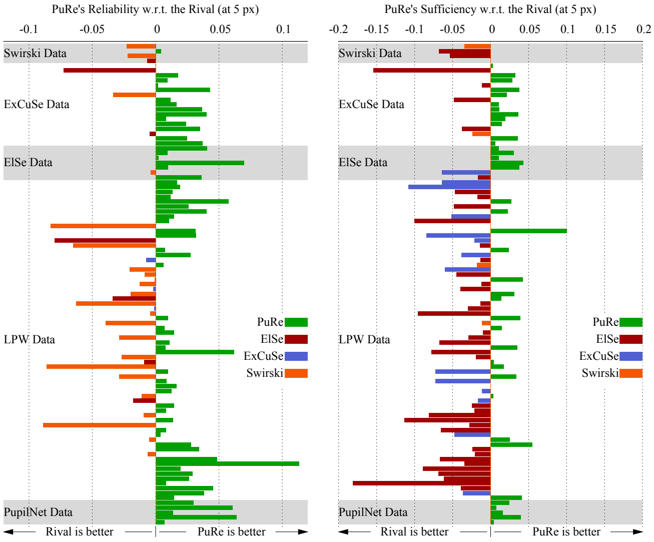


Fig. 15: *PuRe*'s *reliability* (left) and *sufficiency* (right) relative to the rival for each *use case*.

terms of run time, conforming with the slack required for operation with state-of-the-art head-mounted eye trackers. For instance, the Pupil Labs (2017) eye tracker, which provides images at 120 Hz – i.e., a slack of ≈ 8.33 ms. Henceforth, we will use the notation μ for the mean value and σ for the standard deviation. Run time wise, *ExCuSe* was the best performer ($\mu = 2.51, \sigma = 1.11$), followed by *Świrski* ($\mu = 3.77, \sigma = 1.77$), *PuRe* ($\mu = 5.56$, $\sigma = 0.6$), and *ElSe* ($\mu = 6.59$, $\sigma = 0.79$). It is worth noticing that *ElSe* operates on slightly larger images (346 × 260 px) w.r.t. *PuRe* and *ExCuSe* (320 × 240 px). Furthermore, *Świrski* operates on the original image sizes, but its implementation is parallelized using *Intel Thread Building Blocks* (Pheatt, 2008), whereas the other algorithms were not parallelized. In contrast to the algorithmic approaches, Vera-Olmos and Malpica (2017) report run times for their CNN-based approach of ≈ 36 ms and ≈ 40 ms running on a *NVidia Tesla K40 GPU* and a *NVidia GTX 1060 GPU*, respectively. It is worth noticing that these run times are still more than four times larger than the slack required by modern eye trackers and almost one order of magnitude larger than the algorithmic approaches running on a CPU.

5. Discussion

Evaluation results show that our single-method edge-based approach outperformed even two-method approaches (e.g., *ElSe* and *ExCuSe*). However, there are clear (but uncommon) cases when an edge-based approach will not suffice due to lack of edge-information in the image. For instance, extremely blurred images, or if a significant part of the pupil outline is occluded. These challenges might lead *PuRe* to 1) detect only a small part of the pupil outline, which results in a shifted pupil center and an underestimated pupil size, or 2) to fail. In general, *PuRe* has three failure modes, which are depicted in Fig. 17:

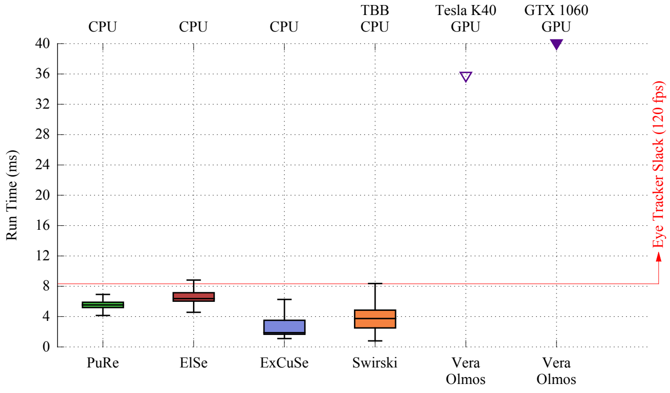


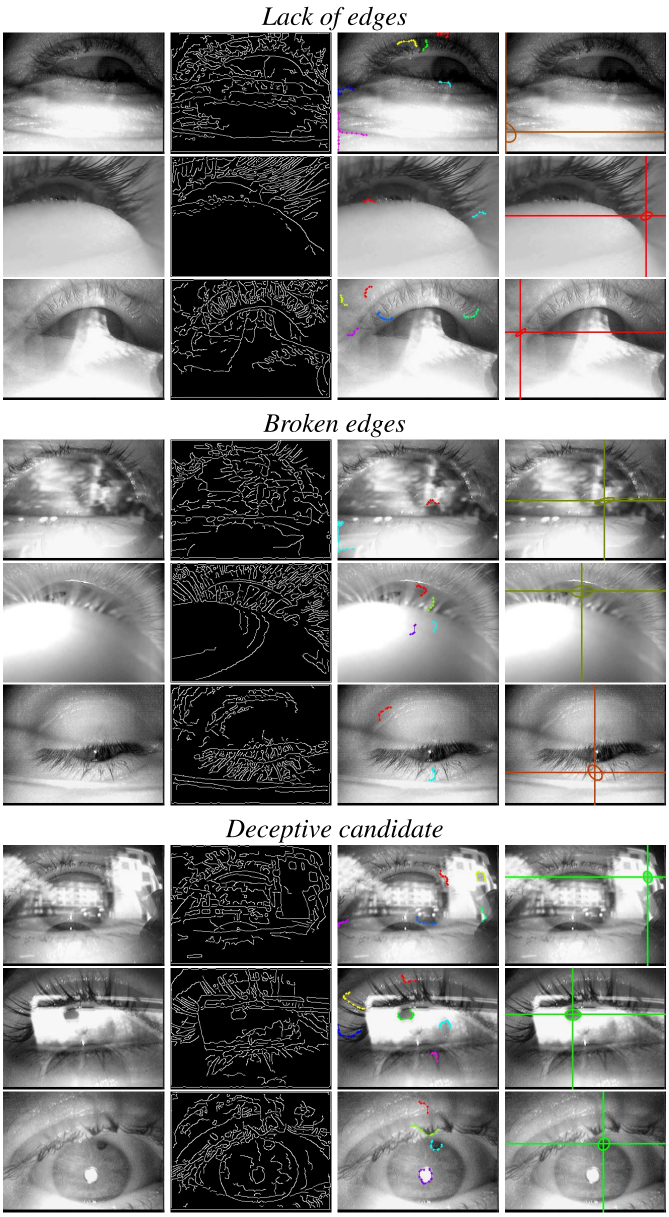
Fig. 16: For *PuRe*, *ElSe*, *ExCuSe*, and *Świrski*: Run time distribution across all images in the *Świrski*, *ExCuSe*, *ElSe*, *LPW*, *PupilNet*, and *Closed-Eyes* data sets. Note that these algorithms were evaluated on a CPU and only *Świrski* was parallelized. For *Vera-Olmos*: Run time as reported in Vera-Olmos and Malpica (2017), which were obtained with parallelized implementations using GPUs.

1. *Lack of edges*: when the pupil outline does not have a contrast strong enough to be detected by the Canny edge detector or is occluded by eyelids / eyelashes / reflections.
2. *Broken edges*: when the pupil outline is broken into smaller parts by eyelids / eyelashes / reflections, which end up removed by the edge segment selection stage.
3. *Deceptive candidate*: when another element in the image *resembles* a pupil more than the pupil itself (according to the definitions of the confidence metric ψ).

It is worth noticing that *PuRe* offers a meaningful confidence measure for the detected pupil, which can be used to identify the great majority of cases in which *PuRe* fails. Following from our analysis in Section 4.2, we recommend a threshold of 0.66 for this confidence measure. Thus, whenever *PuRe* can not find a pupil, an alternative pupil detection method can be employed – e.g., *ElSe*'s fast second step. Nonetheless, care has to be taken not to compromise *specificity* through this second step.

Moreover, there are extreme cases in which pupil detection might not be feasible at all, such as when the bulk of the pupil is occluded due to inadequate eye tracker placement relative to the eye. For instance, *use cases* `LPW/5/6` and `LPW/4/1`, for which the best detection rates were measly 3.45% (by *ExCuSe*) and 14.15% (by *Świrski*), respectively. Sample images throughout these *use cases* are shown in Fig. 18. As can be seen in this figure, in the former not only the eye is out of focus, but there are lenses obstructing most of the pupil, whereas in the latter, the pupil is mostly occluded by the eyelid and eye lashes. In such cases, *PuRe*'s confidence measure provides a quantitative measure of the extend to which it can detect the pupil in current conditions: By observing the ratio of confidence measures above the required threshold during a period[13]. If this ratio is too small, it can be inferred that either the pupil is not visible or *PuRe* can not cope with current conditions. In the

[13] The period should be significantly larger than expected blink durations since the confidence measure is also expected to drop during blinks; in this section we report the ratio for the whole *use case*.



Lack of edges

Broken edges

Deceptive candidate

Fig. 17: Illustrative failure cases for *PuRe*. First column displays the input image, whereas the second column unveils the resulting edges. The third column shows segments remaining after edge segment selection (Section 3.3) using distinct colors per segment. The last column presents the pupil returned by *PuRe*, encoding the confidence measure linearly in the overlay color such that red represents the lowest confidence ($\psi = 0$) and green the highest ($\psi = 1$). Notice that, except for the *deceptive candidates*, the confidence for failures cases is usually low.

former case, the user can be prompted to readjust the position of the eye tracker in real time – this is the case for `LPW/5/6` ($ratio_{th=0.66} = 0.15$) and `LPW/4/1` ($ratio_{th=0.66} = 0.54$). In both cases, the confidence measure ratio is useful for researchers to be aware that the data is not reliable and requires further processing, such as manual annotation. An example of the cases in which adjusting the eye tracker is not likely to improve detection rates is *use case* `LPW/3/16` ($ratio_{th=0.66} = 0.65$), for which reflections cover most of the image as seen in Fig. 18. The best detection rate for this *use case* was 31.95% (by *PuRe*). To further support this claim, we measured the correlation between this confidence-measure ($ratio_{th=0.66}$) and the pupil detection rate, which resulted in a correlation coefficient of 0.88.

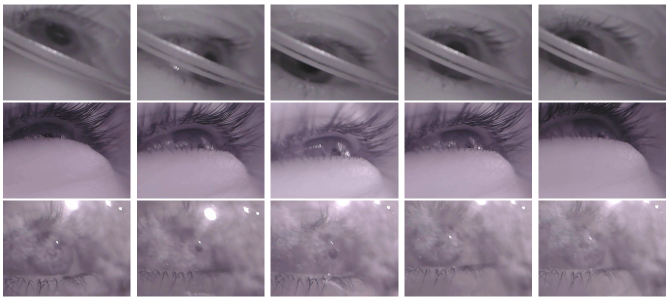


Fig. 18: Extreme cases for pupil detection. For `LPW/5/6` (top row) and `LPW/4/1` (middle row), the eye tracker can be readjusted to improve detection rates. For `LPW/3/16` (bottom row), readjusting the eye tracker is not likely to improve the conditions for pupil detection. In all cases, researchers should be aware that the automatic pupil detection is not reliable. *PuRe*'s confidence measure allows for users to be prompted in real time for adjustments and provides researchers with a quantitative metric for the quality of the pupil detection.

6. Final Remarks

In this paper, we have proposed and evaluated *PuRe*, a novel edge-based algorithm for pupil detection, which significantly improves on the state-of-the-art in terms of *sensitivity*, *precision*, and *specificity* by 5.96, 25.05, and 10.94 percentage points, respectively. For the most challenging data sets, detection rate was improved by more than ten percentage points. *PuRe* operates in real-time for modern eye trackers (at 120 *fps*) and is fully integrated into *EyeRecToo* (Santini et al., 2017a) – an open-source state-of-the-art software for pervasive head-mounted eye tracking. An additional contribution was made in the form of new metrics to evaluate pupil detection algorithms and a data set containing negative samples in its majority. The proposed method and new data set are available at `www.ti.uni-tuebingen.de/perception`.

Envisioned future work includes investigating suitable non-edge-based second steps, and developing a feedback system to prompt users to readjust the eye tracker – to be integrated into *EyeRecToo* (Santini et al., 2017a). Moreover, the focus of this work is on pupil detection – even though we formulated the resulting signal as tracking-by-detection in Section 4.3. Tracking methods face many challenges in eye tracking due to the fast paced changes in illumination, frequent pupil occlusion due to blinks, and substantial amplitude and velocity of saccadic eye movements – as fast as 700 °/s (Baloh et al., 1975). Therefore, a comprehensive evaluation of tracking methods is out of the scope of this paper and better left for future work. Nonetheless, using temporal information (i.e., inter-frame) can greatly benefit detection rates – albeit care has to be taken to track the right element in the image. In this regard, *PuRe*'s confidence metric provides a foundation that may be used in the future to build more robust pupil trackers.

References

Aronson, R., Santini, T., Kübler, T., Kasneci, E., Srinivasa, S., Admoni, H., 2018. Eye-hand behavior in human-robot shared manipulation, in: Proceedings of the 13th Annual ACM/IEEE International Conference on Human Robot Interaction (To appear).

Baloh, R.W., Sills, A.W., Kumley, W.E., Honrubia, V., 1975. Quantitative measurement of saccade amplitude, duration, and velocity. Neurology 25, 1065–1065.

Bashir, F., Porikli, F., 2006. Performance evaluation of object detection and tracking systems, in: Proceedings 9th IEEE International Workshop on PETS, pp. 7–14.

Braunagel, C., Rosenstiel, W., Kasneci, E., 2017. Ready for take-over? a new driver assistance system for an automated classification of driver take-over readiness. IEEE Intelligent Transportation Systems Magazine 9, 10–22.

Bulling, A., Gellersen, H., 2010. Toward mobile eye-based human-computer interaction. IEEE Pervasive Computing 9, 8–12.

Canny, J., 1986. A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence , 679–698.

Čehovin, L., Leonardis, A., Kristan, M., 2016. Visual object tracking performance measures revisited. IEEE Transactions on Image Processing 25, 1261–1274.

Chetlur, S., Woolley, C., Vandermersch, P., Cohen, J., Tran, J., Catanzaro, B., Shelhamer, E., 2014. cudnn: Efficient primitives for deep learning. arXiv preprint arXiv:1410.0759 .

Chu, B.S., Wood, J.M., Collins, M.J., 2010. The effect of presbyopic vision corrections on nighttime driving performance. Investigative ophthalmology & visual science 51, 4861–4866.

Efland, G., Parikh, S., Sanghavi, H., Farooqui, A., 2016. High performance dsp for vision, imaging and neural networks. IEEE Hot Chips 28.

Ergoneers, 2017. Dikablis Glasses Professional. Accessed in 2017-07-26. URL: `http://www.ergoneers.com/en/hardware/eye-tracking/`.

Fitzgibbon, A.W., Fisher, R.B., 1995. A buyer's guide to conic fitting, in: Proceedings of the 6th British Conference on Machine Vision (Vol. 2), BMVA Press, Surrey, UK, UK. pp. 513–522. URL: `http://dl.acm.org/citation.cfm?id=243124.243148`.

Foulsham, T., Walker, E., Kingstone, A., 2011. The where, what and when of gaze allocation in the lab and the natural environment. Vision research 51, 1920–1931.

Frigge, M., Hoaglin, D.C., Iglewicz, B., 1989. Some implementations of the boxplot. The American Statistician 43, 50–54.

Fuhl, W., Geisler, D., Santini, T., Rosenstiel, W., Kasneci, E., 2016a. Evaluation of state-of-the-art pupil detection algorithms on remote eye images, in: Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct, ACM. pp. 1716–1725.

Fuhl, W., Kübler, T., Sippel, K., Rosenstiel, W., Kasneci, E., 2015. Excuse: Robust pupil detection in real-world scenarios, in: International Conference on Computer Analysis of Images and Patterns, Springer. pp. 39–51.

Fuhl, W., Santini, T., Kasneci, G., Kasneci, E., 2016b. Pupilnet: convolutional neural networks for robust pupil detection. arXiv preprint arXiv:1601.04902 .

Fuhl, W., Santini, T.C., Kübler, T., Kasneci, E., 2016c. 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. pp. 123–130.

Fuhl, W., Tonsen, M., Bulling, A., Kasneci, E., 2016d. Pupil detection for head-mounted eye tracking in the wild: an evaluation of the state of the art. Machine Vision and Applications 27, 1275–1288.

Guenter, B., Finch, M., Drucker, S., Tan, D., Snyder, J., 2012. Foveated 3d graphics. ACM Transactions on Graphics (TOG) 31, 164.

Hansen, D.W., Hammoud, R.I., 2007. An improved likelihood model for eye tracking. Computer Vision and Image Understanding 106, 220–230.

Hansen, D.W., Pece, A.E., 2005. Eye tracking in the wild. Computer Vision and Image Understanding 98, 155–181.

Jansen, S., Kingma, H., Peeters, R., 2009. A confidence measure for real-time eye movement detection in video-oculography, in: 13th International Conference on Biomedical Engineering, Springer. pp. 335–339.

Kasneci, E., 2013. Towards the automated recognition of assistance need for drivers with impaired visual field. Ph.D. thesis. Universität Tübingen, Germany.

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 supermarket study. Translational vision science & technology 3, 2–2.

Kristan, M., Matas, J., Leonardis, A., Vojíř, T., Pflugfelder, R., Fernandez, G., Nebehay, G., Porikli, F., Čehovin, L., 2016. A novel performance evaluation methodology for single-target trackers. IEEE transactions on pattern analysis and machine intelligence 38, 2137–2155.

Kübler, T.C., Kasneci, E., Rosenstiel, W., Heister, M., Aehling, K., Nagel, K., Schiefer, U., Papageorgiou, E., 2015. Driving with glaucoma: task performance and gaze movements. Optometry & Vision Science 92, 1037–1046.

Kunjur, J., Sabesan, T., Ilankovan, V., 2006. Anthropometric analysis of eyebrows and eyelids: an inter-racial study. British Journal of Oral and Maxillofacial Surgery 44, 89–93.

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.

Microsoft, 2017. Accessed in 2017-07-26. URL: `https://www.microsoft.com/en-us/hololens`.

Mohammed, G.J., Hong, B.R., Jarjes, A.A., 2012. Accurate pupil features extraction based on new projection function. Computing and Informatics 29, 663–680.

Morimoto, C.H., Mimica, M.R., 2005. Eye gaze tracking techniques for interactive applications. Computer vision and image understanding 98, 4–24.

Oculus, 2017. Accessed in 2017-07-26. URL: `https://www.oculus.com/rift/`.

Pedrotti, M., Lei, S., Dzaack, J., Rötting, M., 2011. A data-driven algorithm for offline pupil signal preprocessing and eyeblink detection in low-speed eye-tracking protocols. Behavior Research Methods 43, 372–383.

Pheatt, C., 2008. Intel® threading building blocks. Journal of Computing Sciences in Colleges 23, 298–298.

Pupil Labs, 2017. Accessed in 2017-07-26. URL: `https://pupil-labs.com/`.

Raffle, H.S., Wang, C.J., 2015. Heads up display. US Patent 9,001,030.

Santini, T., Fuhl, W., Geisler, D., Kasneci, E., 2017a. Eyerectoo: Open-source software for real-time pervasive head-mounted eye-tracking, in: Proceedings of the 12th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications.

Santini, T., Fuhl, W., Kasneci, E., 2017b. Calibme: Fast and unsupervised eye tracker calibration for gaze-based pervasive human-computer interaction, in: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, ACM. pp. 2594–2605.

Santini, T., Fuhl, W., Kübler, T., Kasneci, E., 2016. Bayesian identification of fixations, saccades, and smooth pursuits, in: Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications, ACM. pp. 163–170.

Schmidt, J., Laarousi, R., Stolzmann, W., Karrer-Gauß, K., 2017. Eye blink detection for different driver states in conditionally automated driving and manual driving using eog and a driver camera. Behavior Research Methods , 1–14.

Spector, R., 1990. The pupils, in: Walker HK, Hall WD, H.J. (Ed.), Clinical Methods: The HIstory, Physical, and Laboratory Examinations. Butterworths. chapter 8.

Sugano, Y., Bulling, A., 2015. Self-calibrating head-mounted eye trackers using egocentric visual saliency, in: Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology, ACM. pp. 363–372.

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 and Applications, ACM. pp. 173–176.

Świrski, L., Dodgson, N.A., 2013. A fully-automatic, temporal approach to single camera, glint-free 3d eye model fitting [abstract], in: Proceedings of ECEM 2013.

Teh, C.H., Chin, R.T., 1989. On the detection of dominant points on digital curves. IEEE Transactions on pattern analysis and machine intelligence 11, 859–872.

Tien, T., Pucher, P.H., Sodergren, M.H., Sriskandarajah, K., Yang, G.Z., Darzi, A., 2015. Differences in gaze behaviour of expert and junior surgeons performing open inguinal hernia repair. Surgical endoscopy 29, 405–413.

Tonsen, M., Zhang, X., Sugano, Y., Bulling, A., 2016. Labelled pupils in the wild: a dataset for studying pupil detection in unconstrained environments, in: Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications, ACM. pp. 139–142.

Toussaint, G.T., 1983. Solving geometric problems with the rotating calipers, in: Proc. IEEE Melecon, p. A10.

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.

Vera-Olmos, F., Malpica, N., 2017. Deconvolutional neural network for pupil detection in real-world environments, in: International Work-Conference on the Interplay Between Natural and Artificial Computation, Springer. pp. 223–231.

Vidal, M., Turner, J., Bulling, A., Gellersen, H., 2012. Wearable eye tracking for mental health monitoring. Computer Communications 35, 1306–1311.

Vrzakova, H., Bednarik, R., 2012. Hard lessons learned: mobile eye-tracking in cockpits, in: Proceedings of the 4th Workshop on Eye Gaze in Intelligent Human Machine Interaction, ACM. p. 7.

Wood, J.M., Tyrrell, R.A., Lacherez, P., Black, A.A., 2017. Night-time pedestrian conspicuity: effects of clothing on drivers eye movements. Ophthalmic and physiological optics 37, 184–190.

Zhu, Z., Ji, Q., 2005. Robust real-time eye detection and tracking under variable lighting conditions and various face orientations. Computer Vision and Image Understanding 98, 124–154.