PDIF: Pupil Detection After Isolation and Fitting

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 20, 2020, accepted January 30, 2020, date of publication February 10, 2020, date of current version February 19, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.2973005*

# PDIF: Pupil Detection After Isolation and Fitting

**FEDERICO MANURI[1], ANDREA SANNA[1], AND CHRISTIAN PIO PETRUCCI[2]**
[1]Dipartimento di Automatica e Informatica, Politecnico di Torino, 10129 Turin, Italy
[2]Pay Reply, 10147 Turin, Italy

Corresponding author: Federico Manuri (federico.manuri@polito.it)

**ABSTRACT** Pupil detection plays a key role in eye and gaze video-based tracking algorithms. Various algorithms have been proposed through the years in order to improve the performances or the robustness in real-world scenarios. However, the development of an algorithm which excels in both execution time and pupil detection precision is still an open challenge. This paper presents a novel, feature-based eye-tracking algorithm for pupil detection. Morphological operators are used to remove corneal reflections and to reduce noise in the pupil area prior to the pupil detection step: this solution allows to significantly reduce the computational overhead without lowering the tracking precision. Moreover, a shape validation step is performed after the elliptical fitting and, if the elliptical shape is not detected properly, a set of additional steps is performed to improve the pupil estimation. The proposed solution, Pupil Detection after Isolation and Fitting (PDIF), has been compared with other state-of-the-art tracking algorithms that use morphological operations such as ElSe (Ellipse Selection) and ExCuSe (Exclusive Curve Selector) to evaluate both speed and robustness; the proposed algorithm has been tested over numerous datasets offering different pupil detection challenges. Obtained results show how PDIF provides comparable tracking precision at a significantly lower computational cost compared to ElSe and ExCuSe.

**INDEX TERMS** Pupil detection, eye detection, eye tracking, gaze tracking, image processing, image analysis, human computer interaction.

## I. INTRODUCTION

Eye tracking is the process of measuring the position and movements of the eye: it is a research domain that has been investigated for more than a century because its application can provide significant benefits to human-computer interfaces. Monitoring what the user is looking at can help to achieve a better comprehension of human cognition and intention, enhancing various researches in a plethora of different areas, such as medicine, psychology, marketing, advertising, applications control, autonomous cars and many others.

Information provided by an eye tracker can be used for two different type of interfaces, active and passive. Active interfaces allow users to interact through the use of eye movements [1], e.g. typing on a virtual keyboard by staring at the chosen key [2]. This kind of interfaces is particularly useful for people with disabilities or in situations that do not allow users to interact through more traditional interfaces such as tangible or vocal ones. Passive interfaces follow the user's eyes movements in order to automatically adapt themselves to the user behavior. An example available in virtual reality environments consists in providing the maximum level of details only for objects in the user field of view [3], [4].

Eye tracking technologies can be classified into three categories [5]: eye-attached tracking, optical tracking and electric potential measurement. The first approach uses the movement of an object attached to the eye (e.g., a special contact lens) to measure the eye position. The optical tracking is based on computer vision algorithms, which compute the eye position from an image of the subject (either captured in the infrared or in the visible light spectrum). Finally, the electric potential measurement is based on electrodes positioned around the eyes. The most widely used techniques are currently based on head-mounted optical trackers, mostly due to the advances in the miniaturization of wearable eye-trackers. Moreover, wearable optical trackers are less invasive respect to eye-attached or electric potential ones, and they provide a better accuracy compared to remote video-based eye-tracking. Currently, the market offers a wide plethora of eye trackers ranging from 99 to 40000 Euros [6]–[8]:

The associate editor coordinating the review of this manuscript and approving it for publication was Lefei Zhang.

the state of the art of hardware technologies, combined with the constant improvements of visual analysis algorithms and the spread of mobile and head-mounted devices allow researchers to increase the number of real-world scenarios, overcoming the previous limitations related to studies conducted only in laboratory conditions.

Various algorithms have been proposed through the years in order to improve the performances and the robustness of eye-tracking. However, the development of an algorithm which excels in both execution time and pupil detection precision is still an open challenge. We propose a novel algorithm for eye-tracking named Pupil Detection after Isolation and Fitting (PDIF). The aim of this research is to provide an algorithm which is as reliable as state-of-the-art morphological-based algorithms whereas providing better performances in terms of speed.

The proposed algorithm is inspired by the Starburst [9] algorithm, a morphological operator-based pupil tracker, well-known for its tracking speed; however, only the ray tracing pupil contour detection and the Ransac elliptical fitting steps have been preserved from the original algorithm. The remaining steps have been enhanced by the new proposed techniques to increase the tracking robustness. Moreover, through a set of additional operations aimed to guarantee the correct aspect ratio of the detected pupil, PDIF greatly boosts the tracking precision with respect to Starburst, without introducing noticeable computational overheads. Morphological operators are used to remove corneal reflections and to reduce noise in the pupil area prior to the pupil detection step: this solution allows to significantly reduce the computational overhead without lowering the tracking precision. Moreover, a shape validation step is performed after the elliptical fitting: if the elliptical shape is not detected properly, the algorithm tries to correct the estimation either by repeating the pupil detection step or by using the data from previous frames to improve the pupil estimation. These operations allow the proposed algorithm to significantly improve the tracking performances and make it comparable with state-of-the-art pupil trackers.

Since algorithms such as the Else (Ellipse Selection) algorithm [10] and the ExCuSe (Exclusive Curve Selector) algorithm [11] have been recently proposed in order to improve the Starburst's tracking performance, PDIF has been compared with both the original Starburst and those novel algorithms, which also use morphological operations. The algorithms have been tested considering two different datasets: the ElSe/ExCuSe collection and the Casia collection. Obtained results show how PDIF provides comparable tracking errors at a significantly lower computational cost than ElSe and ExCuSe.

The rest of the paper is organised as follows: the state of the art has been described in detail in Section II. The design and the architecture of the proposed algorithm are presented in Section III. The validation of the PDIF algorithm and its comparison with other solutions are described in Section IV. Conclusions are provided in Section V.

## II. STATE OF THE ART

Eye-tracking algorithms are usually classified as feature-based or model-based [12]. Feature-based algorithms identify features that describe the shape and position of the eye. These algorithms commonly rely on one or more criteria (e.g., thresholds) to establish if a given feature is present or not; by means of the threshold values encoded as parameters, the user can ''tune'' the tracking process. Different features can be chosen depending on the images' spectrum: for example, intensity gradients allow algorithms to detect the pupil contour in infrared spectrum images [13] or the limbus in visible spectrum images [14], [15].

On the other hand, model-based algorithms find the best fitting model that is coherent with the image, such as best-fitting circle [16] or ellipse [17] for the limbus and pupil contour. The model-based approach can provide a more precise estimate of the pupil center than the feature-based approach; it requires to search for a complex parameter space and the search step can return wrong points instead of the real center of the pupil. Hence, a significant cost in terms of computational power might be required to improve the robustness of model-based algorithms.

Hybrid solutions try to overcome the drawbacks of the above mentioned approaches by maintaining the computational cost of feature-based algorithms and the precision of model-based techniques.

In more recent years, with the improvement of machine learning technologies and algorithms, learning-based solutions have been proposed for image-based pupil detection [18], [19]. However, this approach is not comparable with traditional feature-based or model-based solutions due to its intrinsic limitations [20]. Firstly, it requires a training step which is computational and time-consuming. Moreover, since statistics have a high impact on the learning phase, the system has to be trained with a dataset which contains a large variety of distinct challenges in order to obtain a generalized solution. However, since there are limited databases with eye related key points including the eye center annotations, artificial datasets may be necessary to successfully train the system [21]. Overall, if compared with edge selection and ellipse fitting solutions, these methods are generally less accurate, and the runtime is usually comparable to Else and ExCuSe, whereas the average detection rate is improved. Finally, some of these solutions work on images of the whole head/upper body, thus, they suffer from bad environments' illumination conditions, and they provide only an approximated pupil center position (and eventually the eye corners).

The Starburst algorithm proposed by Li *et al.* [9] is considered a valuable solution respect to more recent algorithms in terms of speed. This algorithm sends out rays in multiple directions, then areas with sharp intensity changes along these rays are used as possible pupil border features. The mean position is computed and this step is repeated until convergence. Then, an ellipse is fitted to these features using a RANSAC-based ellipse fitting.

Świrski *et al.* [22] proposed a robust eye tracking algorithm for highly off-axis images, based on an initial approximation of the pupil position through Haar-like features and a refinement step with a modified RANSAC fitting. Other approaches use histogram-based threshold calculation on bright-pupil [23], identify the corneal reflection on IR images using histograms and thresholds-based techniques [24], [25], or rely on symmetric mass center thresholding [26], [27].

The SET algorithm uses a similar approach [28], firstly extracting pupil pixels based on a luminance threshold, then extracting the shape of the thresholded area, and finally comparing it with a sine curve. Valenti et al. proposed an algorithm based on isophotes' curvature estimation and selection of the maximum isocenter as pupil center [29].

Even if many algorithms allow to successfully perform eye tracking under laboratory condition [9], [22]–[27], [30], several studies reported the difficulties arising in eye-tracking applications in natural environments [31]–[35].

Schnipke and Todd [31] summarized a variety of factors that could negatively affect the pupil detection:

- changing illumination conditions, often caused by the subject motion and/or rotation;
- intersection of eyelashes, glasses or contact lenses with the image of the pupil;
- reflections on the pupil due to glasses or contact lenses;
- motion blur;
- the off-axial position of eye camera in head-mounted eye trackers.

Overall, real-world scenario studies regularly report low pupil detection rates and the data collected in such studies have to be manually post-processed [33], [36], [37]. This solution is both laborious and time-consuming and it is also not feasible for real-time applications.

In more recent years, other algorithms have been developed to address robustness in both indoor and outdoor environments such as the ElSe (Ellipse Selection) algorithm [10] and the ExCuSe (Exclusive Curve Selector) algorithm [11]. The main purpose of these researches was to provide robust pupil detection in a wide selection of real-world scenario, whereas previous pupil detection algorithms have been proven reliable almost only in laboratory's tests.

The Exclusive Curve Selector (ExCuSe) was proposed by Fuhl *et al.* [10] in 2015 for real-world eye-tracking applications; this algorithm is based on oriented histograms calculated with the Angular Integral Projection Function [38], used to determine if the image contains reflections. In such a case, the edge image is filtered and the best curve is selected. Otherwise, ellipse estimation is used to refine the coarse pupil center estimation, similarly to Starburst.

The Ellipse Selector (ElSe) algorithm was proposed [11] in 2016: this algorithm, based on edge filtering, ellipse evaluation, and pupil validation, was evaluated through eye-tracking experiments in natural scenarios, proving high detection rates, robustness and fast execution times respect to previous algorithms.

Recently, other researches tried to provide better solutions respect to ElSe and ExCuse but introducing limitations or assumptions that are reasonable only for specific purpose solutions. Abbasi and Khosravi proposed a robust filter-based detection algorithm [39], however, it is highly dependent on the eye conditions, making it difficult to "tune" the parameters controlling the algorithm performance. Moreover, the technique presented in [39] suffers from weak illuminations, reflections and occlusions. Topal *et al.* [40] proposed an adaptive algorithm for precise pupil boundary detection but the tests were performed on a self-made dataset, which does not provide the same number of frames and challenges of traditional benchmarks, such as the ExCuSe/ElSe collection.

Even if these algorithms have been proven to be very reliable and robust, they are quite slower compared to the Starburst algorithm. Since eye-tracking algorithms are usually adopted for real-time applications, it would be valuable to have an algorithm which proves to be both fast as Starburst and reliable as Else or ExCuSe.

The tests presented in this paper prove that the PDIF can estimate the pupil center faster than ExCuSe and Else and with a comparable accuracy; thus, the proposed algorithm can be a valuable instrument to reduce execution times without lowering the robustness.

## III. THE PROPOSED ALGORITHM

The PDIF algorithm has been designed and developed after an accurate analysis of the problems and limitations of concurrent algorithms based on morphological operators. Since Starburst provides the best performances among the other algorithms, it was considered a reasonable starting point for the proposed research. Thus, we analyzed the algorithm trying to understand the reasons for its lack of robustness.

### A. STARBURST ANALYSIS

Starburst is an eye-tracking algorithm able to detect the pupil contour in infrared (IR) spectrum images framed by a head-mounted tracker. Respect to remote trackers, head-mounted ones provide an image limited to the eye area, with a higher or lower resolution depending on the camera quality. Moreover, IR trackers provide a uniform illumination of the eye, which eliminates uncontrolled specular reflections. Even if the algorithm could be adapted for bright-pupil techniques, it has been designed for dark pupil approaches: this means that the eye is illuminated with an off-axis source making the pupil the darkest region in the image. Starburst was first developed in 2005 and it became well-know due to its low computational times and its considerable precision respect to other algorithms of that time such as the Active Shape Models algorithm [41]. However, Starburst is strongly dependent on the size and the quality of the eye images to be processed. Moreover, the algorithm often fails when frames represent sudden pupil movements. Starburst performs well when light conditions are steady and pupil movements are not sudden. Otherwise, the tracking might fail and wrong parts of the eye can be identified as the pupil.

In order to design a solution to these problems, it is necessary to understand which steps of the Starburst algorithm are critical. The first one is the corneal reflections detection and removal. Using the dark-pupil technique, corneal reflections correspond to the brightest regions in the eye image. Starburst makes use of an adaptive thresholding mechanism to detect glints: only values greater than a given threshold are recognized as corneal reflections. The ratio between the area of the largest glint candidate and the other areas is used to identify false corneal reflections. The threshold providing the highest ratio is selected to this end. Then, it is assumed that the intensity profile of a corneal reflection follows a symmetric bivariate Gaussian distribution. Starburst's authors claim to be able to detect up to 98% of corneal reflection extents. However, this method leads to a binary mask that allows to precisely remove the glints in a set of successive frames only if light conditions do not change and shadows do not appear in the frame. Otherwise, the glint detection could be problematic and this will affect the final pupil contour detection. Thus, to provide a robust algorithm, it would be necessary to improve the glint detection in those critical conditions (sudden movements and/or light changes).

Noise can also affect the image, reducing the efficacy of the successive steps of the algorithm and leading to errors. Thus, the second major improvement to obtain a robust algorithm would be to remove all the noise from the image. Noise consists of unusual bright or dark pixels on the image that would mislead the algorithm. The noise can be caused by different elements such as eyelashes, shadows, optical lenses, glasses and lights. To reduce the noise, several algorithms make use of a morphological erosion followed by a kernel expansion to properly remove the glints [42]. However, this technique does not remove the other kinds of noise. Moreover, if the frame is very bright, the light zones on the pupil are enlarged by this operation, creating holes in the image. These holes will increase the chances of errors in the subsequent steps of the algorithm. To avoid this problem, other algorithms combine a kernel dilatation step with a small structuring element and an erosion step with another kernel, bigger than the previous one, as depicted by Rystrom *et al.* [43].

Finally, by analysing the results produced by Starburst, it can be noticed how some wrong detections are related to elongated ellipse shapes. Thus, a third step to enhance the pupil contour recognition would require to improve the elliptical fitting to avoid wrong shapes detected as valid.

### B. NOVEL APPROACH

The flow chart of the proposed algorithm, compared to the steps performed by Starburst, is shown in Fig. 1.

The first step is aimed at reducing the noise in the frame: Starburst was originally developed and tested with a low-cost head-mounted eye tracker that produces noisy images both in terms of shot noise and line noise; the algorithm reduces this kind of noise by applying a 5x5 Gaussian filter with a standard deviation of 2 pixels and a normalization factor applied line by line to shift the mean intensity of the line to the running average derived from previous frames. Since the PDIF algorithm has been tested on frames collections providing high quality images, we propose instead to average each frame by an equalized version of the frame itself. Equalizing the frame allows PDIF to overall improve the image, rebalancing the darkest and lightest areas of the frame. However, this operation can be harmful if performed on an image already providing a good overall contrast since it could alter some pixels, possibly leading the algorithm to errors in the following steps. Thus, instead of a simple equalization, PDIF computes the average between the original frame and its equalized version. This technique is particularly useful for low contrast images obtained by IR cameras because it improves them without introducing noise in already well contrasted ones.

The second step is aimed at removing corneal's reflections due to the IR illuminator. For this step, PDIF applies a morphological closing with a 7x7 elliptical kernel. The choice for the kernel dimension has been based upon the values adopted in similar algorithms which performs morphological operations, such as the one proposed in [44], and the medium glint dimension computed on the collections adopted in the evaluation phase. These are reasonable assumptions since we are working on images provided by a head-mounted IR tracker, thus, the distance between the IR illuminator and the eye should be almost steady, as well as the glint dimension. This step the algorithm is more efficient than Starburst's one in removing glints since it provides good results even with images with shadows or changing light conditions, which have been stated as problematic for Starburst.

The third step of the PDIF algorithm consists of morphological operations aimed at improving the image in order to strengthen the pupil contour detection step. The purpose of this step is to emphasize the pupil area respect to other parts of the eye that could interfere in the recognition process, such as shadow cast by the eyelids. A dilation operation is performed by a 7x7 elliptic kernel, followed by an erosion operation with a slightly larger kernel (12x12). This allows PDIF to detect the region of the pupil with a better precision than Starburst; on the other hand, the size of the pupil after the application of morphological operators is slightly larger than the original one. It is important to notice that at the end of this step the frame used by PDIF for the pupil contour detection is quite different respect to the original one, whereas in Starburst such differences are less noticeable, as shown in Fig. 2. However, as it will be shown in Section IV, this frame's difference does not affect the tracking performance.

Moving forward, the PDIF algorithm performs the same operations as Starburst for the pupil contour detection step. Since the pupil contour usually occupies only a little area of the image and it is not necessary to consider all the pupil contour points to properly estimate the pupil contour, edge detection applied to the entire image or to a region of interest nearby the estimated pupil location is avoided. Instead, the pupil contour is computed by casting a limited number of rays from the center of the pupil, which is estimated manually

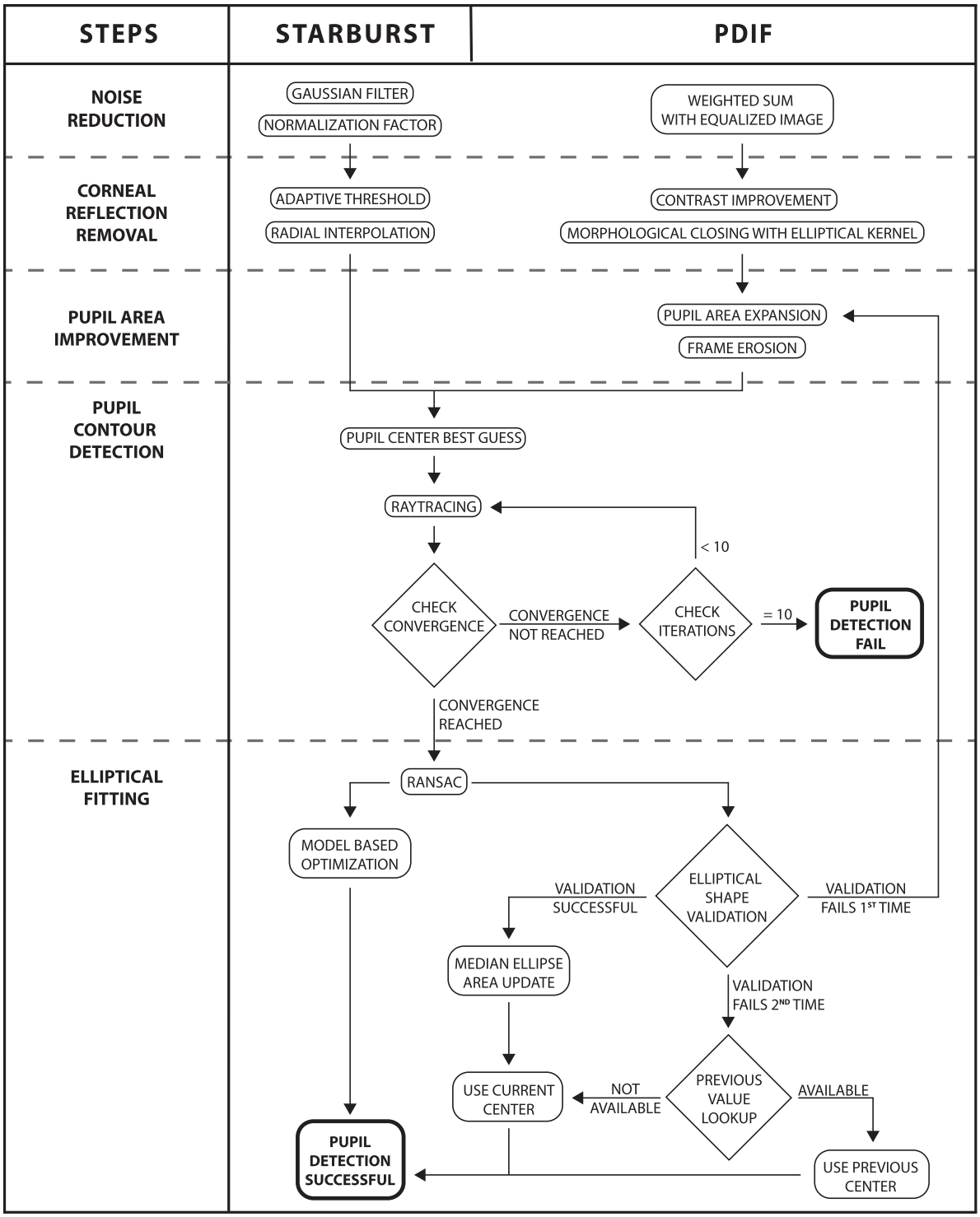



**FIGURE 1.** Flow chart comparison of the PDIF and Starburst algorithms.

for the first frame, whereas the previous detected center is used for the following frames. Moreover, the detection of edges along a limited number of rays extending from the initial set of features toward the starting point is used to improve the robustness against inaccuracy of the starting point. A selectable number of rays is radially traced from the center and the gradient variation is analyzed along each ray (only positive derivatives are considered). The computational time is exponentially dependent on the ray number. For each frame analyzed by the PDIF algorithm, the ellipse is

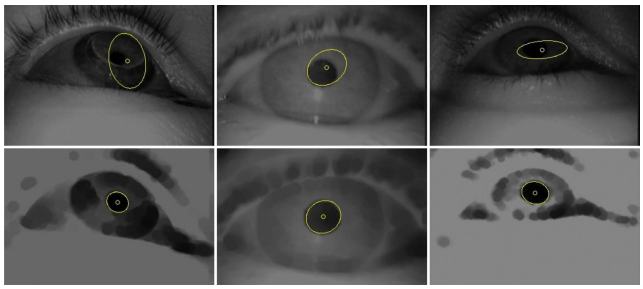


**FIGURE 2.** Pupil detection performed by the Starburst algorithm (first line) versus pupil detection performed by the PDIF algorithm (second line).

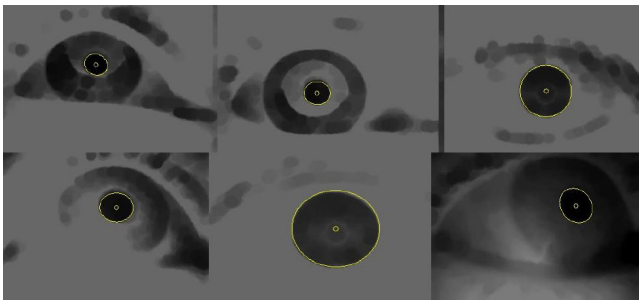


**FIGURE 3.** Pupil detection performed by the PDIF algorithm.

computed tracing 45 rays with a threshold value (the derivatives $\Delta$ along a ray) of 16, which has been proved through preliminary tests on the datasets to be an average good value. The gradient analysis allows to identify a set of edge points along a side of the pupil contour. The edge points are used as starting points to generate a set of rays in the opposite way; this allows to find corresponding edge points on the opposite side of the pupil contour. At least 5 edge points have to be detected in order to draw an ellipse representing the pupil contour; otherwise, the process cannot converge and it is not possible to detect the pupil contour.

Afterwards, the ellipse fitting step takes place: firstly, the Random Sample Consensus (RANSAC) paradigm for model fitting is applied, as in Starburst. Then, in order to improve the ellipse fitting phase, a set of additional steps is performed to check the shape and the dimension of the ellipse. First of all, it is necessary to identify if the given ellipse is elongated or not. This is done evaluating two factors: the ratio between the semi-minor axis and semi-major axis of the ellipse and the ellipse dimension variation over the time. For each dataset in the collection used to test the algorithm, one frame for each type of eyeball rotation respect to the camera have been selected. For each one of these frames, the semi-axises ratio has been computed to evaluate its trend and to define a reference value to discriminate between reasonable ellipses and wrong ones. Likewise, the average ellipse dimension has been computed. As a result, the semi-axises ratio should be higher than 0.4, whereas the ellipse dimension should not be bigger than three times its average size or smaller than one-third of its average size. Otherwise, the ellipse would be too much stretched and will not properly fit the pupil. Thus, if one of these conditions is not met, the pupil contour detection step is repeated, using a threshold value of 25; this value has been tuned through tests on the adopted collections and it should be enough to increase the recognition or even totally overcome tracking errors while minimizing the execution time. After this second ellipse fitting step, if the ellipse is still identified as elongated, the pupil center is approximated using the previous frame center. This is done for a maximum of three frames due to the average time span of eyes' saccades (20-50ms) and fixations (250-300ms). Thus, if the previous frame center has been already used three times, the current center estimation is still considered more accurate than the previous center. Fig. 3 shows six examples of pupil detection performed by the PDIF algorithm.

## IV. ALGORITHM VALIDATION

### A. PRELIMINARY TESTS

Several tests have been carried out to verify the behavior of the PDIF algorithm compared to the original Starburst algorithm. The proposed algorithm consists of two categories of parameters, ''fixed' and ''flexible''. ''Fixed'' parameters allow the algorithm to work properly and lead to very low performances if set to wrong values. ''Flexible'' parameters address variable condition in the source images and should be manually set by the user to further minimize the errors and to enhance the algorithm performances.

The scope of these tests was to identify which parameters pertain to each category and which values should be used as default in order to obtain performances comparable to the original Starburst algorithm. Two datasets of eye images from the ExCuSe and ElSe collection have been selected for these tests, dataset XII and dataset XXIII. These datasets have been chosen because they provide many challenges to be addressed: the first one consists of western eye images which suffer from bad illumination. The second one consists of indoor eastern eye images, with noise produced by eyelids and eyelashes covering the pupil or casting shadows over it.

Taking into account the tests results and the information provided by the state of the art analysis, it was possible to define a set of default values that minimize both the error rate computed on the total collection of images and the average error rate computed for each dataset. In particular, the used default parameters are: dilation kernel 7x7, erosion kernel 12x12, scattered rays 45, derivative $\Delta$ along the rays 16 (then 25), ellipse aspect ratio grater than 0.4, ellipse fitting dimension bigger than three times the average ellipse size and smaller than one-third of the average ellipse size.

### B. PERFORMANCE EVALUATION

The performances of the PDIF algorithm have been evaluated with publicly available datasets populated by infrared images. Tests have been carried out on the collection provided by the authors of the ExCuSe and ElSe algorithms, which consist of images of a single eye with a low resolution (384x288 pixels).

The CASIA collection, made available from the Chinese Academy of Sciences Institute of Automation, has also been used for testing the PDIF algorithm: this collection consists of two distinct datasets, each containing 239 images of a single eye, one for the right eye and the other one for the left eye. This collection consist of high definition's images that have been obtained using a set of IR illuminators displaced on a circular perimeter. Both collections consist only of images representing the ocular area instead of the whole face because all the algorithms involved in the tests belong to the head mounted, eye-tracking category, and they have been designed to work best with this type of images. These collections allowed to test the PDIF algorithm in many conditions, since each dataset contains different challenges in terms of light condition, camera position respect to eyes, eyes typology, presence of reflections or other negative conditions.

### C. METRICS

Different metrics have been taken into account to assess the PDIF performances. First, the algorithm's performance in term of execution times (in milliseconds) for each frame has been evaluated: this is considerably important since one of the main goal of the proposed algorithm is to be feasible for a real-time application usage. Thus, we expect the execution time to be comparable to the Starburst algorithm, and overall considerably faster than the other considered trackers. Secondly, since the PDIF algorithm wants to overcome the precision limitations of the Starburst algorithm, it should prove to be more robust in challenging conditions. Thus, the following metrics have been considered to properly evaluate its precision and robustness: first, the detection rate for different pixel errors based on the Euclidean distance between the hand-labeled ground-truth and the pupil center; the curve representing this metric should ideally converge towards a step function. Moreover, the average distance in pixel between the real pupil center and the estimated one has been considered: to obtain a robust eye-tracking, the system should provide a low average distance. The percentage of errors in the pupil center estimation greater than 30 pixels was also considered, since this value could be considered as a meaningful threshold (for the frames' resolution of the considered datasets) between almost accurate detections and completely wrong ones. Another important metric that was computed is the standard deviation from the real center: this value is relevant since it provides a statistical evaluation of the algorithm, representing the dispersion or variability of the obtained results. Finally, for the PDIF algorithm, the number of right and wrong approximations of the pupil center when performing a second ellipse computation and the number of right and wrong approximations when referring to the previous pupil center estimation were also computed. Overall, the Starburst algorithm has been chosen as a reference, even if more recent algorithms perform better in terms of precision. The different metrics have been computed as delta percentage (either positive or negative) respect to the Starburst's values.

### D. TESTS

In order to compare the performances of the PDIF algorithm with state-of-the-art solutions, three other algorithms have been tested processing the same datasets: the Starburst [9] algorithm was used without any changes in the parameter settings and the starting location was set to the center of the image. The ExCuSe [10] and ElSe [11] algorithms were used with the parameter settings provided by the authors. In order to provide a fair comparison, the PDIF algorithm was used with the default parameters (PDIF Default, see Section IV-A) to all datasets. However, the algorithm has been designed to be customizable in terms of contrast and kernel size for the morphological operations through a simple graphic user interface (GUI). In order to evaluate the bounds of the proposed algorithms, each dataset has been tested customizing the PDIF parameters to maximize the performances (PDIF Optimal). The results will show that even if some metrics benefit from this tuning, PDIF Defaults still performs better in others, thus proving that the default version is a meaningful solution respect to its competitors.

### E. EXCUSE/ELSE COLLECTION

This collection comprehends twenty-four different datasets that have been proposed by the ExCuSe/ElSe algorithms' authors for pupil detection performance testing. This collection has been made freely accessible by the research group of the Eberhard Karls Universität Tübingen, which proposed the two algorithms. Each dataset counts a variable number of images and poses different challenges. The first group consists of fifteen datasets that were recorded with a head-mounted camera during an on-road driving experiment [36]. The second one counts eight datasets which provide images from a search task in a supermarket and were not recorded for pupil detection studies [35]. The last one consists of two datasets that were obtained from indoor experiments with Asian subjects and provides challenges related to eyelids and eyelashes covering the pupil or casting shadows over it and reflection on eyeglasses.

Overall, we computed a total of 5 metrics for each of the 24 dataset; since most of them share the same challenges, we decided to report the test results only for a subset which meaningfully represents the performances (and limitations) of the proposed algorithm respect to the others. The chosen datasets are representative of the whole collection both in term of challenges and obtained results. For the first group, whose main challenges consist of bad illumination and reflections, test results for dataset number II and IV are reported. For the second one, which provided a more heterogeneous set of challenges, comprehending pupil at image border, bad illumination, eyelashes and shifted contact lenses, test results for dataset number X, XII, XIII, XIV, XV and XVII are reported. For the last group, whose main challenges consist of eastern eyes, shadows cast by eyelids and eyelashes or pupil covered by eyelids or eyelashes, test results for dataset XXIII are reported.

**TABLE 1.** Column Starburst shows the average execution time for each dataset in milliseconds. Columns PDIF, Excuse and ElSe show the average execution time for each dataset both in milliseconds and in percentage delta from Starburst's values.

| Dataset | Starburst | PDIF | ExCuSe | ElSe |
|---|---|---|---|---|
| II | 8,91 ms | **9,40 ms** | 20,42 ms | 23,73 ms |
| IV | 3,97 ms | **6,00 ms** | 9,77 ms | 20,37 ms |
| X | 3,51 ms | **7,27 ms** | 18,20 ms | 19,87 ms |
| XII | 3,13 ms | **6,54 ms** | 10,11 ms | 19,56 ms |
| XIII | 2,98 ms | **4,10 ms** | 10,42 ms | 20,38 ms |
| XIV | 2,41 ms | **7,19 ms** | 19,06 ms | 19,82 ms |
| XV | 2,96 ms | **5,99 ms** | 13,56 ms | 21,28 ms |
| XVII | 12,31 ms | **11,24 ms** | 23,60 ms | 19,65 ms |
| XXIII | 1,63 ms | **6,14 ms** | 6,34 ms | 24,00 ms |
| **Average** | 4,65 ms | **7,10 ms** | 14,61 ms | 20,96 ms |
| II | 8,91 ms | **5,50%** | 129,18% | 166,33% |
| IV | 3,97 ms | **51,13%** | 146,10% | 413,10% |
| X | 3,51 ms | **107,12%** | 418,52% | 466,10% |
| XII | 3,13 ms | **108,95%** | 223,00% | 524,92% |
| XIII | 2,98 ms | **37,58%** | 249,66% | 583,89% |
| XIV | 2,41 ms | **198,34%** | 690,87% | 722,41% |
| XV | 2,96 ms | **102,36%** | 358,11% | 618,92% |
| XVII | 12,31 ms | **-8,69%** | 91,71% | 59,63% |
| XXIII | 1,63 ms | **276,69%** | 288,96% | 1372,39% |
| **Average** | 4,65 ms | **52,76%** | 214,47% | 351,23% |

**TABLE 2.** Average distance between the real pupil center and the estimated one, in pixel for Starburst and in percentage delta from the Starburst's value for the other algorithms.

| Dataset | Starburst | PDIF (D) | PDIF (O) | ExCuSe | ElSe |
|---|---|---|---|---|---|
| II | 76,70 | -65,80% | -75,91% | -62,29% | **-89,57%** |
| IV | 20,08 | -62,85% | **-65,89%** | -28,44% | -18,18% |
| X | 50,52 | -64,15% | **-64,65%** | -64,35% | -42,95% |
| XII | 32,76 | **-80,04%** | -79,55% | -62,27% | 30,68% |
| XIII | 62,89 | **-68,91%** | -59,90% | -40,90% | -63,62% |
| XIV | 28,46 | -29,34% | **-75,76%** | -24,24% | 18,24% |
| XV | 16,78 | 10,97% | **-18,30%** | 103,28% | 62,75% |
| XVII | 178,80 | -92,21% | **-97,47%** | -86,07% | -85,48% |
| XXIII | 20,04 | **-81,19%** | -80,39% | -54,74% | -79,09% |
| **Average** | 54,12 | -72,42% | **-78,61%** | -58,84% | -56,87% |

Table 1 shows the average execution time for each dataset in milliseconds for the Starburst algorithm. For PDIF, ExCuSe and Else, the average execution time for each dataset both in milliseconds and in percentage delta from the Starburst's values is reported. The last line shows the average over the whole group of datasets. As expected, the Starburst algorithm is the fastest one. However, the PDIF always performs better than ExCuSe and Else. On average, the PDIF is only 52,76% slower than Starburst, whereas the proposed algorithm performs numerous extra steps to improve the pupil detection precision. Moreover, the test results shows that for some challenges the PDIF execution time is almost comparable to Starburst. Overall, these results prove that the PDIF algorithm can be valuable in real-time scenarios in terms of execution times.

Table 2 shows the average distance between the real pupil center and the estimated one, in pixel for Starburst and in percentage delta from the Starburst's value for the other algorithms. The last line shows the average over the whole group of datasets. Overall, the PDIF algorithm provides the most significant improvement respect to the other algorithms, with an average distance more than 70% lower than Starburst on average. In the worst case scenario, the average distance reduction is comparable to that of ExCuSe (dataset X). It is relevant to observe that PDIF (Optimal) obtains better result than PDIF (Default) most of the time, but depending on the dataset's challenges it can even perform slightly worse. These results prove that tuning the parameters can improve the values for some metrics at the expense of reducing the performances of others.

**TABLE 3.** Percentage of errors in the pupil center estimation greater than 30 pixels.

| Dataset | Starburst | PDIF (D) | PDIF (O) | ExCuSe | ElSe |
|---|---|---|---|---|---|
| II | 72,08% | 21,19% | 13,47% | 16,63% | **3,96%** |
| IV | 16,80% | **3,43%** | 4,37% | 9,08% | 15,78% |
| X | 58,93% | 20,48% | **7,38%** | **7,38%** | 13,93% |
| XII | 27,67% | **2,67%** | **2,67%** | 5,53% | 15,46% |
| XIII | 47,25% | 17,52% | 14,05% | 17,11% | **6,92%** |
| XIV | 19,19% | 16,63% | **2,99%** | 5,76% | 13,86% |
| XV | 13,22% | 15,70% | 11,02% | 17,63% | **10,19%** |
| XVII | 96,28% | 11,52% | **0,74%** | 4,09% | 9,76% |
| XXIII | 12,42% | **0,63%** | **0,63%** | 4,09% | 2,83% |
| **Average** | 40,43% | 12,20% | **6,37%** | 9,70% | 10,30% |

**TABLE 4.** Standard deviation for the errors in the pupil center estimation in pixels for Starburst and the percentage delta from the Starburst's value for the other algorithms.

| Dataset | Starburst | PDIF (D) | PDIF (O) | ExCuSe | ElSe |
|---|---|---|---|---|---|
| II | 216,48 | -88,93% | -82,97% | -57,51% | **-92,98%** |
| IV | 39,99 | **-67,59%** | -63,17% | 49,06% | -12,75% |
| X | 54,41 | **-48,41%** | -1,62% | 3,44% | 31,21% |
| XII | 60,5 | **-73,02%** | -69,21% | -30,71% | 60,51% |
| XIII | 74,24 | **-65,67%** | -16,49% | 12,53% | -10,24% |
| XIV | 48,39 | -65,26% | **-77,66%** | 130,85% | 60,59% |
| XV | 30,44 | -4,73% | **-19,91%** | 141,16% | 123,65% |
| XVII | 786,42 | -97,27% | **-99,40%** | -73,31% | -90,29% |
| XXIII | 2,09 | **103,83%** | 230,62% | 1569,38% | 811,96% |
| **Average** | 145,88 | **-86,42%** | -82,29% | -41,87% | -59,91% |

Table 3 shows the percentage of errors in the pupil center estimation greater than 30 pixels (i.e., when the pupil is not recognized). The last line shows the average over the whole group of datasets. Overall, the PDIF (Default) obtains an average value slightly lower than ExCuSe and ElSe. However, it can even outmatch the other algorithms depending on the dataset challenges (datasets IV, XII and XXIII). As expected, the PDIF (Optimal) outmatches the other algorithms most of the time, otherwise obtaining values comparable to ExCuSe and ElSe.

Table 4 shows the standard deviation for the errors in the pupil center estimation in pixels for Starburst and the percentage delta from the Starburst's value for the other algorithms. The last line shows the average over the whole group of datasets. Except for dataset II, the PDIF (Default) always outmatches ExCuSe and Else, performing on average better than Starburst by 86,42% and the PDIF (Optimal) obtains the best overall averages, with values considerably lower than the other algorithms. Since standard deviation is strictly related to the average distance between the real pupil center and the estimated one, it is relevant to observe that PDIF (Optimal) can obtain better result than PDIF (Default) sometimes, but

**TABLE 5.** Number of right and wrong approximations of the pupil center when performing a second ellipse computation and number of right and wrong approximations when referring to the previous pupil center estimation.

| | Second pupil detection approximation | | Previous frame centre approximation | |
|---|---|---|---|---|
| **Dataset** | Right | Wrong | Right | Wrong |
| II | 76,60% | 17,74% | 64,56% | 35,44% |
| IV | 84,09% | 18,26% | 74,60% | 25,40% |
| X | 28,57% | 17,86% | 63,89% | 36,11% |
| XII | 50,00% | 22,22% | 58,33% | 41,67% |
| XIII | 0,00% | 7,41% | 54,35% | 45,65% |
| XIV | 100,00% | 0,00% | 62,50% | 37,50% |
| XV | 50,00% | 23,53% | 54,17% | 45,83% |
| XVII | 0,00% | 0,00% | 100,00% | 0,00% |
| XXIII | 0,00% | 0,00% | 0,00% | 0,00% |
| **TOTAL** | 76,42% | 16,45% | 65,80% | 34,20% |

depending on the dataset's challenges it can perform even worse.

Fig. 4 shows the charts representing the detection rate precision in term of pixel errors. Each chart refers to a different dataset of the ExCuSe/ElSe Collection. The algorithms are represented with different colors. Since the curve representing this ratio should ideally converge towards a step function, a good algorithm should both rapidly rise and be able to reach a high value. PDIF obtains the worst result in dataset XIV, performing as Starburst. It achieves better results in dataset II and X, performing better than Starburst but worse than ExCuSe and ElSe. In dataset XIII and XVII, PDIF performs slightly worse than ElSe and ExCuSe, whereas its performance is comparable to these two algorithms for the dataset XV. Finally, it performs as well or even better than ExCuSe and ElSe in dataset IV, XIV and XXIII (in this last case performing as well as PDIF Optimal). PDIF (Optimal) instead usually obtains results comparable to ExCuSe and ElSe (datasets II, X, XII, XIII and XV) whereas it can even outperform them in some cases (datasets IV, XIV and XVII).

Finally, table 5 shows the number of right and wrong approximations when a second ellipse is computed and the number of right and wrong approximations when referring to the previous pupil center estimation. Since these steps are peculiar of the PDIF algorithm, it is not possible to compare them with the other algorithms. However, a relevant outcome is that when a second pupil contour detection step is required, the final approximation is often correct (76,42% of the times). Moreover, when the algorithm needs to rely on the pupil center estimated for a previous center, 65,80% of the times it leads to a right approximation.

### F. CASIA COLLECTION

The Casia collection provides images at a higher definition than the previous one and with a resolution of 320x280 pixels. The eyes are framed from 239 different subjects and the collection contain 2664 frames, half for the left eye and half for the right one. Since this collection was created using IR illuminators, it represents a valuable benchmark to evaluate if the proposed algorithm can effectively avoid the problems related to the glints created by the illuminators over the pupil.

**TABLE 6.** Column Starburst shows the average execution time for each dataset in milliseconds. Columns PDIF, Excuse and ElSe show the average execution time for each dataset both in milliseconds and in percentage delta from Starburst's values.

| | AVERAGE EXECUTION TIME | | | |
|---|---|---|---|---|
| **Dataset** | Starburst | PDIF | ExCuSe | ElSe |
| CASIA L | 8,53 ms | **8,91 ms** | 10,73 ms | 20,30 ms |
| CASIA R | 7,67 ms | **10,44 ms** | 10,97 ms | 20,22 ms |
| **Average** | 8,10 ms | **9,68 ms** | 10,85 ms | 20,26 ms |
| CASIA L | 8,53 ms | **4,45%** | 25,79% | 137,98% |
| CASIA R | 7,67 ms | **36,11%** | 43,02% | 163,62% |
| **Average** | 8,1 ms | **19,44%** | 33,95% | 150,12% |

**TABLE 7.** Average distance between the real pupil center and the estimated one, in pixel for Starburst and in percentage delta from the Starburst's value for the other algorithms. Percentage of errors in the pupil center estimation greater than 30 pixels. Standard deviation for the errors in the pupil center estimation in pixels for Starburst and the percentage delta from the Starburst's value for the other algorithms.

| | AVERAGE DISTANCE FROM THE CENTRE | | | | |
|---|---|---|---|---|---|
| **Dataset** | Starburst | PDIF (D) | PDIF (O) | ExCuSe | ElSe |
| CASIA L | 18,32 | -77,35% | **-81,93%** | -72,27% | -78,88% |
| CASIA R | 16,16 | -77,66% | **-81,06%** | -67,39% | -77,78% |
| **Average** | 17,24 | -77,49% | **-81,53%** | -69,98% | -78,36% |
| | ERRORS >30 PIXEL | | | | |
| **Dataset** | Starburst | PDIF (D) | PDIF (O) | ExCuSe | ElSe |
| CASIA L | 16,29% | 0,53% | **0,08%** | 1,43% | 0,30% |
| CASIA R | 13,62% | **0,08%** | **0,08%** | 1,99% | 0,23% |
| **Average** | 14,96% | 0,31% | **0,08%** | 1,71% | 0,27% |
| | STANDARD DEVIATION FROM CENTRE | | | | |
| **Dataset** | Starburst | PDIF (D) | PDIF (O) | ExCuSe | ElSe |
| CASIA L | 54,83 | -91,61% | **-95,24%** | -49,24% | -90,61% |
| CASIA R | 34,57 | -89,41% | **-94,24%** | -20,28% | -90,60% |
| **Average** | 44,7 | -90,76% | **-94,85%** | -38,04% | -90,60% |

As for the previous collection, Starburst is the fastest algorithm. However, as depicted in Table 6, PDIF is only 19,44% slower than Starburst on average and is faster than ExCuSe and ElSe.

The detection rate precision of the PDIF (Default) algorithm in term of pixel errors (Fig. 5) is far better than Starburst and it is comparable to the other trackers. PDIF (Optimal) always performs better than the other algorithms.

The low values of both the average error in term of pupil center estimation and the standard deviation prove that the PDIF algorithm is well-designed for this kind of collection (Table 7). Moreover, PDIF outmatches the other algorithms also in terms of center estimation error greater than 30 pixels. Finally, the algorithm never required to perform an additional processing through a second ellipse estimation or a comparison with the previous frame pupil center.

### G. RESULTS ANALYSIS

Starburst is a popular and robust eye-tracking algorithm but its limitations when light conditions are not steady and/or pupil movements are sudden between successive frames provided the opportunity to propose a novel, feature-based eye-tracking algorithm. More recent algorithms (ExCuSe, ElSe) overcome Starburst's precision limitations but introduce a

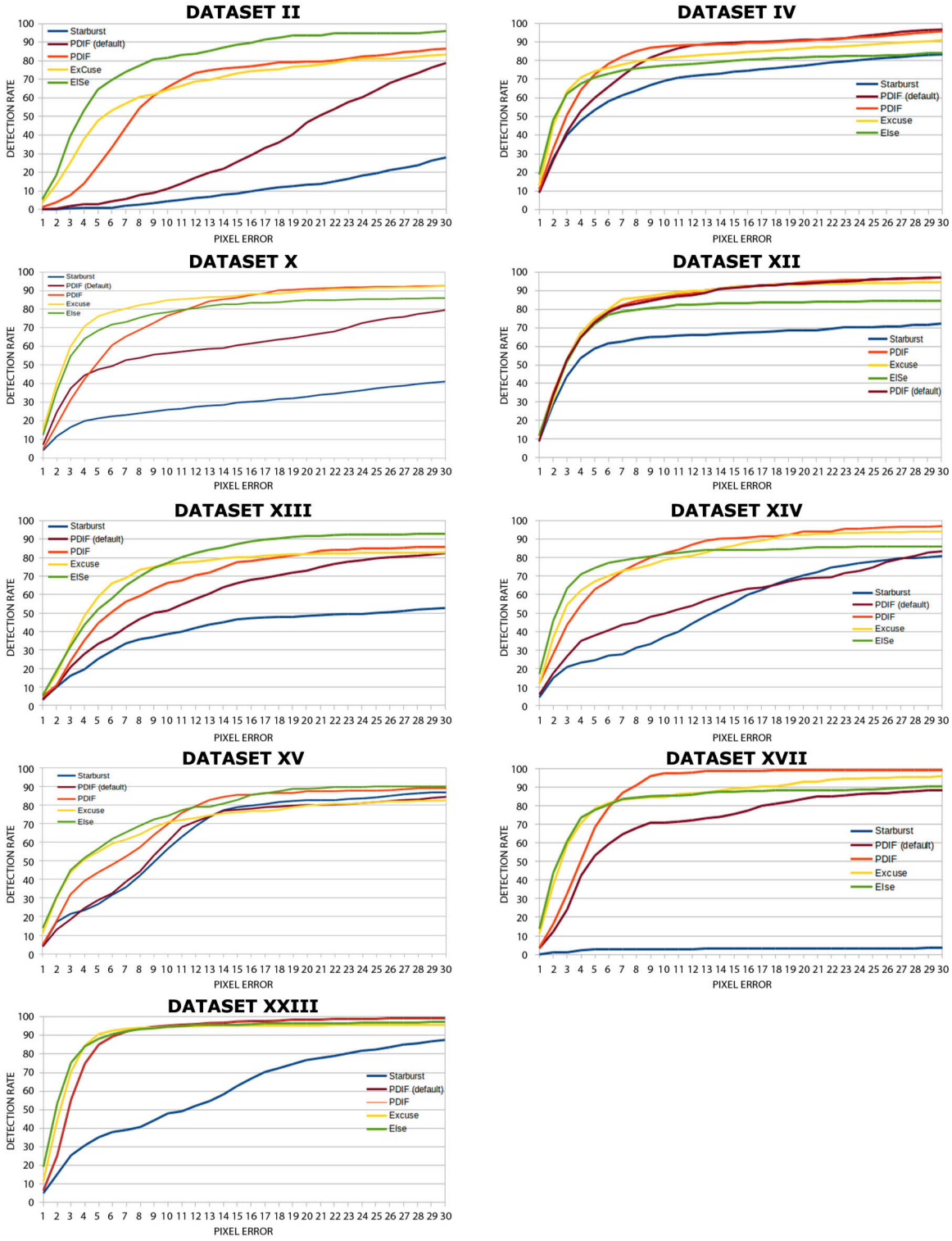



**FIGURE 4.** Charts representing the precision in term of pixel errors. Each chart refers to a different dataset of the ExCuSe/ElSe Collection. The algorithms are represented with different colours.

computational delay that makes them unsuitable for real-time applications.

The PDIF algorithm introduces a set of operations which greatly boost its precision compared to Starburst, proving to be not only reliable but also considerably fast. PDIF results to be suitable for real-time applications, as it is, on the average, faster than ElSe and ExCuSe in terms of execution times. Considering the robustness of the proposed algorithm, in the

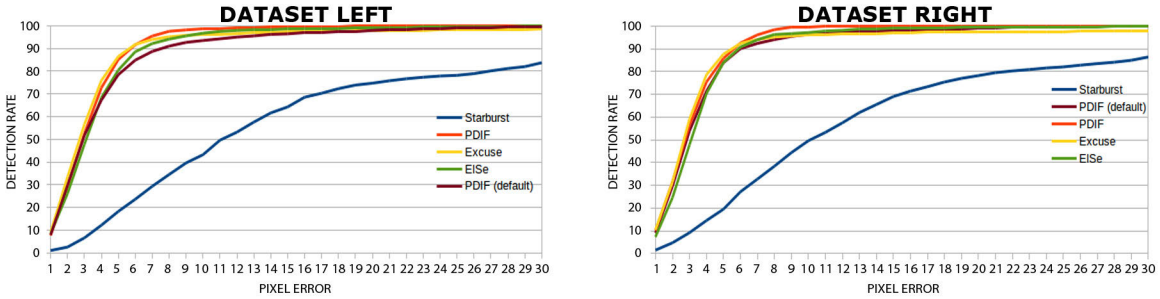



**FIGURE 5.** Charts representing the detection rate precision in term of pixel errors. Each chart refers to a different dataset of the ExCuSe/ElSe Collection. The algorithms are represented with different colours. The left chart refers to the Casia Left database, the right one to the Casia Right database. The algorithms are represented with different colours.

first collection used for the tests, the ElSe and ExCuSe algorithms proved to be valuable competitors due to their high precision and low average distance error, performing overall slightly better than PDIF. However, the errors committed by the PDIF are minor, whereas those committed by ElSe or ExCuSe are substantial: this leads to an average error and a standard deviation that rewards PDIF. When considering the CASIA dataset, the proposed algorithm is almost equivalent to ExCuSe and Else.

## V. CONCLUSION

This paper presents a novel, feature-based eye-tracking algorithm, which tries to provide good performances in real-world scenario both in terms of execution speed and robustness. Morphological operators are used to remove corneal reflections and reduce noise in the frame prior to the pupil detection step, allowing the proposed algorithm to significantly reduce the computational overhead without lowering the tracking precision. Moreover, a shape validation step is performed after the elliptical fitting, thus, even if the elliptical shape is not detected properly, the tracking algorithm tries to correct the estimation. The proposed solution, Pupil Detection after Isolation and Fitting (PDIF), has been compared with other tracking algorithms that use morphological operations such as ElSe and ExCuSe, to evaluate both speed and robustness. Obtained results show how PDIF provides comparable tracking errors at a significantly lower computational cost.

In the future, we would like to test the performances of the proposed algorithm with trackers which provide images of the whole head, e.g. not wearable trackers. In such cases, the frames of the ocular area should present a higher level of noise and a lower contrast respect to head-mounted trackers. However, since the morphological steps used in the PDIF algorithm specifically address this type of problems, we expect it to perform well. Another possible development may involve machine learning approaches to automatically tune the contrast and kernel size parameters: this could improve the PDIF performances such as for the PDIF Optimal, without requiring the user to manually tune the parameters. Moreover, it would be interesting to further improve the elliptical shape validation step trying to predict the pupil shape depending on the previous frames and on the pupil position respect to the eye (e.g. if the pupil moves toward one side, we would expect it to get an elongated shape).

## REFERENCES

[1] R. J. K. Jacob, "The use of eye movements in human-computer interaction techniques: What you look at is what you get," *ACM Trans. Inf. Syst.*, vol. 9, pp. 152–169, Apr. 1991.

[2] P. Majaranta and K.-J. Räihä, "Twenty years of eye typing: Systems and design issues," in *Proc. Symp. Eye tracking Res. Appl.*, 2002, pp. 15–22.

[3] D. J. Parkhurst and E. Niebur, "Variable-resolution displays: A theoretical, practical, and behavioral evaluation," *Hum. Factors*, vol. 44, no. 4, pp. 611–629, Dec. 2002.

[4] D. Parkhurst and E. Niebur, "A feasibility test for perceptually adaptive level of detail rendering on desktop systems," in *Proc. 1st Symp. Appl. Perception Graph. Vis.*, 2004, pp. 49–56.

[5] L. R. Young and D. Sheena, "Survey of eye movement recording methods," *Behavior Res. Methods Instrum.*, vol. 7, no. 5, pp. 397–429, Sep. 1975.

[6] *The Eye Tribe Tracker, an Eye Tracking System that can Calculate the Location Where a Person is Looking by Means of Information Extracted From Person's Face and Eyes*. Accessed: Oct. 15, 2019. [Online]. Available: https://theeyetribe.com/theeyetribe.com/about/index.html

[7] *The Eyelink 1000 Plus, an Eye Tracking System Produced by SR Research, an Accurate Video-Based Eye Tracker, Sampling Binocularly at up to 2000 Hz*. Accessed: Oct. 15, 2019. [Online]. Available: https://www.sr-research.com/products/eyelink-1000-plus/

[8] *The Eyegaze Edge is a Tablet With a Special Eye Tracking Camera Mounted Below the Screen Which can Track One of the User's Eyes*. Accessed: Oct. 15, 2019. [Online]. Available: https://eyegaze.com/products/eyegaze-edge/

[9] D. Li, D. Winfield, and D. Parkhurst, "Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR) - Workshops*, Jan. 2006, p. 79.

[10] W. Fuhl, T. Kübler, K. Sippel, W. Rosenstiel, and E. Kasneci, "Excuse: Robust pupil detection in real-world scenarios," in *Proc. Int. Conf. Comput. Anal. Images Patterns*. Cham, Switzerland: Springer, 2015, pp. 39–51.

[11] W. Fuhl, T. C. Santini, T. Kübler, and E. Kasneci, "Else: Ellipse selection for robust pupil detection in real-world environments," in *Proc. 9th Biennial ACM Symp. Eye Tracking Res. Appl.*, 2016, pp. 123–130.

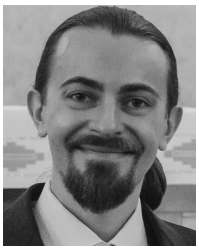
[12] D. Li and D. J. Parkhurst, "Starburst: A robust algorithm for video-based eye tracking," *Elselvier Sci.*, vol. 6, 2005.

[13] T. Ohno, N. Mukawa, and A. Yoshikawa, "Freegaze: A gaze tracking system for everyday gaze interaction," in *Proc. Symp. Eye tracking Res. Appl.*, 2002, pp. 125–132.

[14] J. Zhu and J. Yang, "Subpixel eye gaze tracking," in *Proc. IEEE Int. Conf. Autom. Face Gesture Recognit.*, May 2002, pp. 131–136.

[15] X. L. Brolly and J. B. Mulligan, "Implicit calibration of a remote gaze tracker," in *Proc. Conf. Comput. Vis. Pattern Recognit. Workshop (CVPRW)*, 2004, p. 134.

[16] J. Daugman, "High confidence visual recognition of persons by a test of statistical independence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 11, pp. 1148–1161, Nov. 1993.
[17] K. Nishino and S. K. Nayar, "Eyes for relighting," in *Proc. ACM Trans. Graph.*, vol. 23, 2004, pp. 704–711.
[18] A. Larumbe, R. Cabeza, and A. Villanueva, "Supervised descent method (SDM) applied to accurate pupil detection in off-the-shelf eye tracking systems," in *Proc. ACM Symp. Eye Tracking Res. Appl.*, 2018, p. 7.
[19] F. Vera-Olmos, E. Pardo, H. Melero, and N. Malpica, "DeepEye: Deep convolutional network for pupil detection in real environments," *Integr. Comput.-Aided Eng.*, vol. 26, no. 1, pp. 85–95, 2019.
[20] W. Fuhl, D. Geisler, T. Santini, T. Appel, W. Rosenstiel, and E. Kasneci, "CBF: Circular binary features for robust and real-time pupil center detection," in *Proc. ACM Symp. Eye Tracking Res. Appl.*, 2018, p. 8.
[21] C. Gou, H. Zhang, K. Wang, F.-Y. Wang, and Q. Ji, "Cascade learning from adversarial synthetic images for accurate pupil detection," *Pattern Recognit.*, vol. 88, pp. 584–594, Apr. 2019.
[22] L. Świrski, A. Bulling, and N. Dodgson, "Robust real-time pupil tracking in highly off-axis images," in *Proc. Symp. Eye Tracking Res. Appl.*, 2012, pp. 173–176.
[23] S. Goni, J. Echeto, A. Villanueva, and R. Cabeza, "Robust algorithm for pupil-glint vector detection in a video-oculography eyetracking system," in *Proc. 17th Int. Conf. Pattern Recognit. (ICPR)*, vol. 4, 2004, pp. 941–944.
[24] A. Keil, G. Albuquerque, K. Berger, and M. A. Magnor, "Real-time gaze tracking with a consumer-grade video camera," in *Proc. WSCG Full Papers*, 2010, pp. 129–134.
[25] L. Lin, L. Pan, L. Wei, and L. Yu, "A robust and accurate detection of pupil images," in *Proc. 3rd Int. Conf. Biomed. Eng. Informat. (BMEI)*, vol. 1, 2010, pp. 70–74.
[26] X. Long, O. K. Tonguz, and A. Kiderman, "A high speed eye tracking system with robust pupil center estimation algorithm," in *Proc. 29th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBS)*, Aug. 2007, pp. 3331–3334.
[27] A. Peréz, M. L. Córdoba, A. Garcia, R. Méndez, M. Munoz, J. L. Pedraza, and F. Sanchez, "A precise eye-gaze detection and tracking system," in *Proc. WSCG Posters Papers*, 2003, pp. 105–108.
[28] A.-H. Javadi, Z. Hakimi, M. Barati, V. Walsh, and L. Tcheang, "SET: A pupil detection method using sinusoidal approximation," *Frontiers Neuroeng.*, vol. 8, p. 4, Apr. 2015.
[29] R. Valenti and T. Gevers, "Accurate eye center location through invariant isocentric patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1785–1798, Sep. 2012.
[30] D. Zhu, S. T. Moore, and T. Raphan, "Robust pupil center detection using a curvature algorithm," *Comput. Methods Programs Biomed.*, vol. 59, no. 3, pp. 145–157, Jun. 1999.
[31] S. K. Schnipke and M. W. Todd, "Trials and tribulations of using an eye-tracking system," in *Proc. Extended Abstracts Hum. Factors Comput. Syst. (CHI)*, 2000, pp. 273–274.
[32] E. Kasneci, "Towards the automated recognition of assistance need for drivers with impaired visual field," *Wilhelmstr*, vol. 32, p. 72074, 2013.
[33] X. Liu, F. Xu, and K. Fujimura, "Real-time eye detection and tracking for driver observation under various light conditions," in *Proc. IEEE Intell. Vehicle Symp.*, vol. 2, Jun. 2002, pp. 344–351.
[34] E. Kasneci, K. Sippel, M. Heister, K. Aehling, W. Rosenstiel, U. Schiefer, and E. Papageorgiou, "Homonymous visual field loss and its impact on visual exploration: A supermarket study," *Trans. Vis. Sci. Tech.*, vol. 3, no. 6, p. 2, Oct. 2014.
[35] K. Sippel, E. Kasneci, K. Aehling, M. Heister, W. Rosenstiel, U. Schiefer, and E. Papageorgiou, "Binocular glaucomatous visual field loss and its impact on visual exploration—A supermarket study," *PLoS ONE*, vol. 9, no. 8, Aug. 2014, Art. no. e106089.
[36] E. Kasneci, K. Sippel, K. Aehling, M. Heister, W. Rosenstiel, U. Schiefer, and E. Papageorgiou, "Driving with binocular visual field loss? A study on a supervised on-road parcours with simultaneous eye and head tracking," *PLoS ONE*, vol. 9, no. 2, Feb. 2014, Art. no. e87470.
[37] S. Trösterer, A. Meschtscherjakov, D. Wilfinger, and M. Tscheligi, "Eye tracking in the car: Challenges in a dual-task scenario on a test track," in *Proc. Adjunct 6th Int. Conf. Automotive User Inter. Interact. Veh. Appl.*, 2014, pp. 1–6.
[38] G. J. Mohammed, B. R. Hong, and A. A. Jarjes, "Accurate pupil features extraction based on new projection function," *Comput. Informat.*, vol. 29, no. 4, pp. 663–680, 2012.
[39] M. Abbasi and M. R. Khosravi, "A robust and accurate particle filter-based pupil detection method for big datasets of eye video," *J. Grid Comput.*, pp. 1–21, Dec. 2019.
[40] C. Topal, H. I. Cakir, and C. Akinlar, "An adaptive algorithm for precise pupil boundary detection using entropy of contour gradients," Dept. Comput. Eng., Anadolu Univ., Eskişehir, Turkey, 2017.
[41] A. Abhyankar and S. Schuckers, "Active shape models for effective iris segmentation," *Proc. SPIE*, vol. 6202, Apr. 2006, Art. no. 62020H.
[42] Y. Ebisawa, "Improved video-based eye-gaze detection method," *IEEE Trans. Instrum. Meas.*, vol. 47, no. 4, pp. 948–955, 1998.
[43] L. R. Rystrom, P. L. Katz, R. M. Haralick, and C. J. Eggen, "Morphological algorithm development case study: Detection of shapes in low-contrast gray-scale images with replacement and clutter noise," *Proc. SPIE*, vol. 1658, pp. 76–94, Apr. 1992.
[44] M. A. Abdullah, S. Dlay, and W. Woo, "Fast and accurate pupil isolation based on morphology and active contour," in *Proc. 4th Int. Conf. Signal, Image Process. Appl.*, 2014, pp. 6–7.



**FEDERICO MANURI** received the B.Sc., M.Sc., and Ph.D. degrees in computer engineering from the Politecnico di Torino, Italy, in 2008, 2011, and 2017, respectively. He is currently a Postdoctoral Research Assistant with the Dipartimento di Automatica e Informatica, Politecnico di Torino. His research interests include human–machine interaction, computer graphics, and augmented reality.

**ANDREA SANNA** received the M.Sc. degree in electronic engineering and the Ph.D. degree in computer engineering from the Politecnico di Torino, Italy, in 1993 and 1997, respectively. He is currently an Associate Professor with the Dipartimento di Automatica e Informatica, Politecnico di Torino. He has authored or coauthored several articles in the areas of human–machine interaction, computer graphics, scientific visualization, augmented and virtual reality, distributed computing, and computational geometry. He is also a member of several scientific committees and a Reviewer of journals and conferences. He is also a Senior Member of the ACM.

**CHRISTIAN PIO PETRUCCI** received the B.Sc. and M.Sc. degrees in computer engineering from the Politecnico di Torino, Italy, in 2014 and 2016, respectively. He is currently a Java Developer, a Database Administrator, and a Computer System Administrator with Pay Reply, Turin, Italy.