Eye Pupil Detection Using OpenCV And Python

Abdullah Abdullah Matr: 1635047
Johnal Dsouza Matr: 1650259

Abstract

Fast and Robust detection is the prerequisite of many ubiquitous eye tracking applications in real world environments such as Gaze Estimation, Human-Computer Interaction and Advanced Driver Assistance. Accurate pupil detection is crucial as errors in detection can degrade the performance in its respective applications. High frequency noise caused due to presence of eyelashes can be mistaken for a part of pupil, hence reducing the accuracy of the detection model. Our paper proposes a method for robust real time pupil detections using computer vision techniques in OpenCV and Python using both Gaussian and Median blurring reducing the false edges due to eyelids, eyelashes hair while capturing the image, thus preserving only the pupil and increasing the accuracy and detection speed which can be integrated into embedded architectures.

1 Introduction

Non-invasive eye-tracking and real-time eye-pupil detection has revealed an enormous research opportunity, particularly in the fields of human computer interface, virtual reality, and medical diagnostics[1]. It assists in the understanding and study of human behavior in terms of emotions and mental wellness, tracking of one's physical and psychological health as well as keeping track of cognitive loads[1],[2]. Many eye detection methods have evolved in recent years as a result of developments in eye-pupil detection and tracking, assisting doctors in diabetic retinopathy screening and detecting eye diseases[3][4]. Eye detection and tracking is an essential research topic in computer vision and image processing, which leads to various applications such as gaze tracking, biometric identification for unlocking the smartphone[1][4], and in automobiles for advanced driver assistance[5] and drivers' fatique management[1].

We argue in this paper an optimized pupil detection pipeline. We proposed an efficient, accurate and faster pupil detection pipeline comprising of both Median and Gaussian blurring for effectively reducing noise, improved edge detection using Canny algorithm, resulting in overall faster processing speed and contributing to more accurate and efficient pupil detection and lastly filtering detected Contours based on certain area and circularity criterias and only the best circular contour based on circularity is considered as a potential pupil candidate.The dataset on which the pupil detection algorithm is performed was collected by a self wearable non-invasive infrared camera embedded device[2] in the form of a video. The video was converted to frames consisting of approximate 1500 frames. This paper discuss the problem of widening(dilate) or shortening(contract) of eyes pupil because of the effect of environment brightness and darkness on the eye similar to the mechanism of camera lens[4] and also provides solutions to the difficulties encountered owing to eyebrows and eyelashes[4]. The circularity algorithm effectively handles the contraction and dilatation of the eye.

In the next part, we will discuss the most recent and effective state-of-the-art approaches for eye recognition and tracking, as well as the methodology utilized in detail for each method, followed by experiments and outcomes in terms of accuracy, speed, and precision. Finally, we describe the conclusion, which highlights the primary output of the results obtained.

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

2 Related work

A well known state-of-the-art pupil detection pipeline ElSe(Ellipse Selection) is proposed by Wolfgang Fuhl[6]. Several phases are included in the pupil detection pipeline employed in this paper. First, the input image is converted to grayscale. Then, 10% of the image's border region is removed to lessen the impression of eyeglass frames. After that, the image is adjusted to improve the contrast. The image is then subjected to a Canny edge filter to detect edges. Edge connections are deleted to guarantee that the pupil's surrounding edge is not harmed. Straightness, inner intensity value, elliptic qualities, the ability of fitting an ellipse to it, and a pupil plausibility check are all used to evaluate connected edges. Similar to edge detection technique by Wolfgang FUhl [6], PuRe(Pupil Reconstruction) algorithm also uses a Canny edge operator for edge detection, followed by a morphological approach to thin and straighten edges and break up orthogonal connections[5]. This results in an image with unconnected and thinned edge segments. PuRe then uses a novel edge segment selection and conditional segment combination schemes to detect the pupil[5]. Finally, PuRe includes a confidence measure for the detected pupil.

Ankur Raj[1] employs the most essential image processing eye pupil algorithms with a non-invasive embedded platform for pupil detection. It adds another advanced technique to the same use of canny edge detection. It crops the image to an area of interest (ROI), converts the image to grayscale, and reduces noise with a median blur filter. The image's edges are then obtained using Canny edge detection, and the intensity of the eyelashes is reduced using a morphological open operation. The generated image is then processed to detect and segment the pupil contour using an elliptical fitting approach. With a run-time of 54ms at 480x640 and 23ms at 240x320, the method is tuned for real-time performance.

3 Dataset

The dataset consists of two video files with approximately 1500 frames of the pupil of two different persons using a non-invasive infrared camera. The video files were passed to the detection algorithm, where center position of the detected pupil was stored and the video files were later hand-annotated by interactively drawing a rectangle using the mouse covering the area of the pupil, thus generating ground truth data for training and evaluating pupil detection algorithms. The top left and bottom left coordinate positions are stored for further analysis.

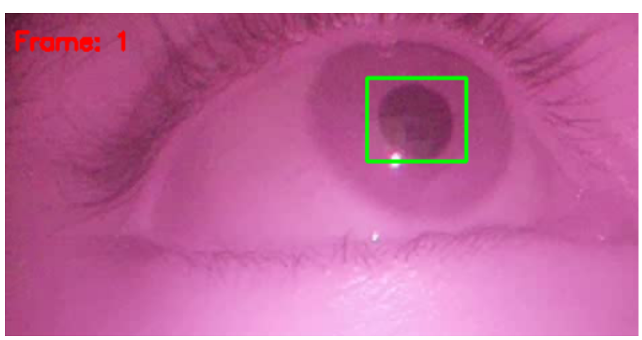

Figure 1: Example of annotation performed on dataset

4 Methodology

Preprocessing, Circular Contour detection and selecting the best circular contour comprise the pupil detection steps. The image operations used in the proposed technique is as described in the Fig2 below

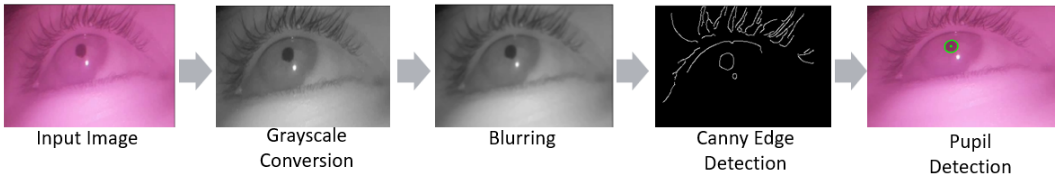

Figure 2: Pupil detection pipeline

4.1 Preprocessing

4.1.1 Selecting Region of Interest (ROI) and Grayscale Conversion

The images obtained from the infrared camera have a resolution of 480x640. As a larger image will result in larger computation time, therefore it is necessary to crop the images, creating a required Region of Interest (ROI). The ROI is defined as the upper-left portion of the frame with a resolution of 500x350 where the eye is expected to be located. Necessary considerations need to be taken such that the ROI includes the entire eyes, not leaving any part missing. As edge detection algorithms primarily rely on the intensity gradients between neighbouring pixels, hence necessary conversion from B-G-R to grayscale is necessary to distinguish the pupil from the iris and other eye structures, at the same time ensuring faster and simpler computational computing compared to coloured images[7].

4.1.2 Blurring

To increase the accuracy of pupil detection, it is necessary to remove the presence of eyelashes. The presence of eyelashes creates high frequency noise, which can be mistaken for a part of pupil, hence reducing the accuracy of the detection model[6]. In the proposed algorithm we have used both Gaussian and Median blurring, the resulting blurring results in removal of major part edges of the eyelashes, preserving only the pupil. In the proposed algorithm, Median blurring was performed with a kernel size of 5 on the grayscale image, followed by Gaussian blurring on the median blurring image with kernel size of 7. In Gaussian blur, a border type parameter `'cv2.BORDER_REPLICATE'` was used, which ensures that the border pixels will be replicated from the edge of the image, further preserving the image's content at the borders, thus producing a smooth and continuous result. In Fig3 shows Canny edge detected images, that by using both Gaussian and Median blur reduces the presence of eyelashes, thus increasing the accuracy of detection of pupil.

Figure 3: Canny edge detected images. Left: Only Median blur image. Middle: Only Gaussian blur image. Right: Median + Gaussian blur image

4.1.3 Canny Edge Detection and Contour Extraction

Canny Edge Detection is a widely used edge detection technique, that by incorporating the gradient information constructs a binary edge map of the eye, which is necessary for localization[8]. The double thresholding in Canny Edge Detection helps in identifying potential edges and suppressing weaker and noise-related edges. A morphological closing operation was performed on the edges to close gaps and fill small holes in the edges. In Contour Extraction, the contours are extracted from the closed edges with `'cv2.CHAIN_APPROX_TC89_KCOS'` method, which is the process of representing a contour with a reduced number of points while preserving its shape and minimizing loss of important information[5].

4.2 Circular Contour detection

The proposed algorithm filters out circular contours (pupil candidate) based on their area and circumference. For each contour, the convex hull is calculated to approximate its shape. Contours with an area greater than a particular threshold value and with a circumference less than or equal to a particular threshold value, which are found by brute force are considered as potential pupil candidates. The circularity of each contour is computed using the formula:

$$\text{circularity} = \frac{4 \times \pi \times \text{area}}{\text{circumference}^2}$$

The formula calculates a value that quantifies how close the contour resembles a perfect circle. A perfect circle has a circularity of 1, and as the shape deviates from a circle, the circularity value approaches zero. In the proposed algorithm, the circularity is used to filter out contours that are not circular enough, helping to identify the best circular contour representing the pupil.

4.3 Best Circular Contour Selection

From the filtered pupil candidates, the algorithm selects the best circular contour that represents the pupil. For each pupil candidate, its corresponding circularity is computed. The pupil with the lowest circularity is considered the best fit for the eye's pupil. The center of the pupil is calculated by finding the centroid of the best-fit contour using moments, which will be later used for analysis. The algorithm draws the best-fit ellipse around the detected pupil.

5 Results

Our method was evaluated on two new hand-labeled datasets consisting of approx. 1500 images. For analysis of our algorithm, we have checked whether the detected centers fall within bounding rectangles regions, which can help evaluate the accuracy of the pupil detection algorithm, provide valuable information about its performance, and provides valuable insights for further improvements if needed. The achieved results from the two datasets are mentioned in the Table1 given below. The algorithm achieved an overall reasonable performance in accuracy and recall rate and high performance in precision, demonstrating its ability to effectively detect pupil in our datasets. The algorithm does fail when the pupil is at the extreme top right corner of the eye, reflection from a bright light source and fast motion of pupil (motion blur). The measured run-time was 3.2 ms per image on an Intel i7-10750H CPU (2.60GHz)

Table 1: Results of Detection on two datasets

	Dataset1	**Dataset2**
Total frames	1412	1580
Correctly detected pupil frames	1135	1213
Incorrectly detected pupil frames	24	42
No detection of pupil in frame	253	325
Accuracy	80.38%	76.77%
Incorrect Detection Rate	1.70%	2.66%
Missed Detection Rate	17.92%	20.57%
Precision	97.93%	96.65%
Recall	81.77%	78.87%
F1 Score	81.77%	86.87%
Average FPS	347.70	363.13

6 Conclusion

In this paper, the technique of using both Gaussian and Median blurring reduces the false edges due to eyelids, eyelashes hair while capturing the image, thus preserving only the pupil increasing the accuracy of detection of pupil. The suggested technique proves to be an efficient solution for detecting pupil features for various ubiquitous eye tracking applications due to its high precision, reasonable accuracy, and faster speed.

References

[1] Ankur Raj, Diwas Bhattarai, and Kristof Van Laerhoven. An embedded and real-time pupil detection pipeline. *arXiv preprint arXiv:2302.14098*, 2023.

[2] Li Ding, Jack Terwilliger, Aishni Parab, Meng Wang, Lex Fridman, Bruce Mehler, and Bryan Reimer. Clera: A unified model for joint cognitive load and eye region analysis in the wild. *ACM Transactions on Computer-Human Interaction*, 2023.

[3] Nasro Min-Allah, Farmanullah Jan, and Saleh Alrashed. Pupil detection schemes in human eye: a review. *Multimedia Systems*, 27(4):753–777, 2021.

[4] D Gokulavani and J Mohana. A performance analysis of different detection techniques of human eye pupils. In *2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS)*, pages 824–827. IEEE, 2023.

[5] Thiago Santini, Wolfgang Fuhl, and Enkelejda Kasneci. Pure: Robust pupil detection for real-time pervasive eye tracking. *Computer Vision and Image Understanding*, 170:40–50, 2018.

[6] Wolfgang Fuhl, Thiago C Santini, Thomas Kübler, and Enkelejda Kasneci. Else: Ellipse selection for robust pupil detection in real-world environments. In *Proceedings of the ninth biennial ACM symposium on eye tracking research & applications*, pages 123–130, 2016.

[7] Rafael C Gonzalez. *Digital image processing*. Pearson education india, 2009.

[8] Milena BP Carneiro, Antônio CP Veiga, Fernando C Castro, Edna L Flôres, and Gilberto A Carrijo. Processing the segmentation stage of an iris recognition system through evolutionary algorithm. *Journal of Communication and Information Systems*, 24(1), 2009.