NIST Interagency Report
NIST IR 8516

An Open Source Iris Segmentation Algorithm for Operational Images

George W. Quinn
James R. Matey

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8516

NIST Interagency Report
NIST IR 8516

An Open Source Iris Segmentation Algorithm for Operational Images

George W. Quinn
James R. Matey

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8516

March 2024

U.S. Department of Commerce
Gina M. Raimondo, Secretary

National Institute of Standards and Technology
Laurie E. Locascio, NIST Director and Under Secretary of Commerce for Standards and Technology

Publication History
Approved by the NIST Editorial Review Board on 2024-03-06

How to cite this NIST Technical Series Publication:
George W. Quinn, James R. Matey (2024) An Open Source Iris Segmentation Algorithm for Operational Images. (National Institute of Standards and Technology, Gaithersburg, MD), NIST IR 8516. https://doi.org/10.6028/NIST.IR.8516

NIST Author ORCID iDs
George W. Quinn: 0000-0001-8414-8392
James R. Matey: 0000-0002-0508-785X

Contact Information
gw@nist.gov

Abstract

This document describes a hough-based approach to localizing the iris boundaries in iris images. The proposed algorithm starts by applying a series of morphological operations to enhance the salience of the iris boundaries while mitigating the impact of noise. A circular hough transform is then used to detect the pupil boundary. Finally, the fit is optimized via a gradient ascent algorithm that maximizes an objective reward function that quantifies the curve's goodness-of-fit. The efficacy of the proposed algorithm is demonstrated over two datasets, Notre Dame 0405 and the OPS 4 iris dataset used for principle performance testing in the IREX 10 ongoing evaluation. For the first dataset, the algorithm was able to correctly localize the boundaries in 831 out of 837 images, or 99.3% of the time, making it competitive with other open source algorithms. For the second, the algorithm was able to correctly localize the boundaries 99.2% of the time. The algorithm's source code, which utilizes the standard OpenCV computer vision library, is free for developers and universities to download and use.

Keywords

biometrics; IREX; iris localization; iris recognition; iris segmentation.

Table of Contents

1. Introduction 1
2. Proposed Method 1
   2.1. Image Preprocessing 2
      2.1.1. Specular Highlight Removal 2
      2.1.2. Eyelash Mitigation 3
      2.1.3. Contrast Enhancement 3
      2.1.4. Gaussian Blur 3
      2.1.5. Gradient Information 3
   2.2. Pupil Localization 3
   2.3. Limbus Localization 5
3. Experimental Results 5
4. Summary 7
5. Notes 7
References 7

List of Tables

List of Figures

Fig. 1. Iris Localization Steps 2
Fig. 2. Examples from ND-0405 of correctly located iris boundaries. 5
Fig. 3. Examples of improperly located iris boundaries 6

1. Introduction

Iris recognition has seen widespread use in recent years. The NYPD uses the iris to ensure that arrestees are properly arraigned in connection to their case [1]. India's Aadhaar scheme [2] uses iris recognition on a national scale to fairly allocate resources to hundreds of millions of Indian residents. The United Arab Emirate's (UAE's) border-crossing system [3] screens individuals at ports of entry, prohibiting those previously expelled from reentering the country. The FBI is encouraging federal, state, and local law enforcement agencies to submit iris samples acquired during booking to its Next Generation Identification (NGI) system, which currently contains 2.5 million enrollments [4].

Traditionally, the first step in the identification process is to locate the iris within the image. This involves locating the inner pupil boundary and the outer limbus. The pupil boundary outlines the pupil aperture while the limbus marks the transition from the iris to the sclera. Special circumstances aside (e.g., the eye condition known as Coloboma), the two boundaries tend to form an approximately annular shape, although a portion of the upper and lower iris tend to be occluded by the eyelids. For many popular matching algorithms, proper localization of the boundaries is critical to correctly identifying the individual. That said, some of the newer deep-learning approaches to iris matching [5, 6] do not require explicit localization of the iris boundaries.

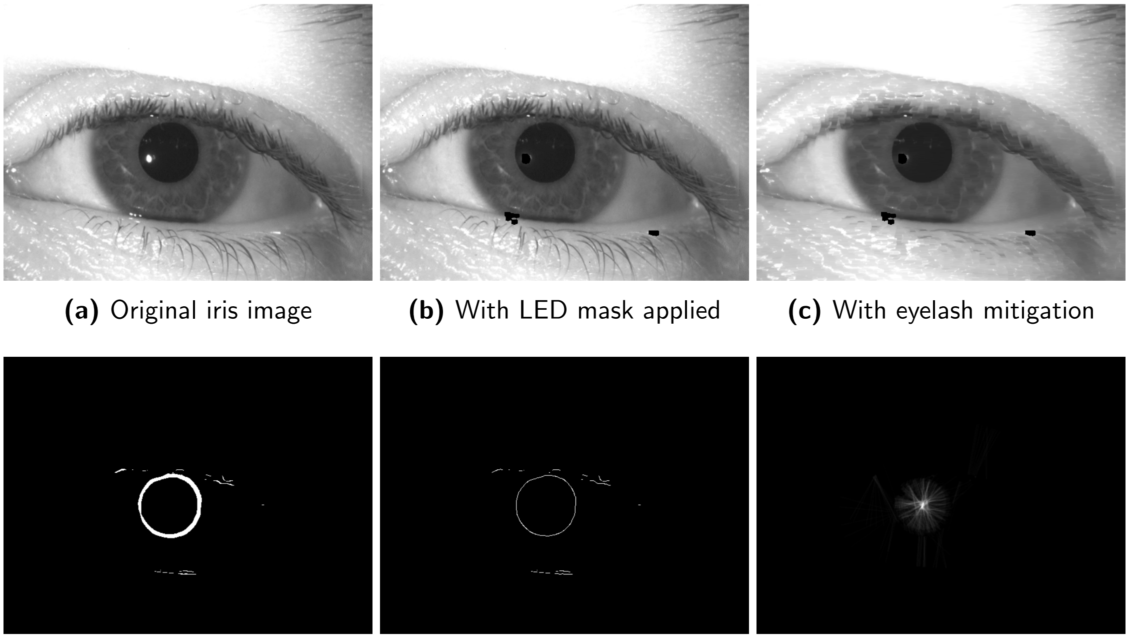
Localization of the iris boundaries is likely to be more difficult if the image was acquired in a less strictly controlled environment, where the person may have been squinting or looking off-axis at the moment of capture. Motion blur, occlusion from the eyelids and eyelashes, and off-axis gazes, are all problems that commonly affect iris images. A robust iris boundary localization algorithm should be able to overcome such obstacles in all but the most extreme cases.

This document describes a new approach to localizing the iris boundaries in iris images. The algorithm roughly builds on Bendale *et. al.* [7] in that it utilizes a similar hough transform to localize the pupil. The implementation is open-source and was developed by NIST to collect and report statistics on its sequestered iris test datasets [8]. The core source files consist of about 500 lines of code and utilize the OpenCV Library [9] to perform morphological operations and to find optimal parameter values that minimize arbitrary cost functions.

The remainder of this paper is organized as follows. Section 2 details the proposed method. Second 3 presents experimental results over two datasets, and Section 4 summarizes the work.

2. Proposed Method

Prior to searching for the iris, the image is preprocessed to both increase the salience of the iris boundaries and to reduce any noise that may have been introduced by a poor-quality



(a) Original iris image **(b)** With LED mask applied **(c)** With eyelash mitigation

(d) Hough Mask **(e)** Skeletonized Mask **(f)** Hough Space

Fig. 1. The original iris image (a), the image with the LED mask applied (b), the image with eyelash mitigation (c), the hough mask (d), the skeletonized hough mask (e), and a 2D representation of the hough space ultimately used to localize the pupil.

capture. A hough transform is then applied to identify the approximate radius and center of the pupil. The fit is then optimized using an iterative gradient ascent algorithm. Finally, localization of the limbus is accomplished by again utilizing the gradient ascent algorithm with a similar maximization function.

2.1. Image Preprocessing

2.1.1. Specular Highlight Removal

Iris cameras use Light Emitting Diodes (LEDs) to illuminate the iris. These LEDs frequently introduce specular highlights to the acquired images. Although the specular pattern can, and often does, vary from camera to camera, the individual LEDs typically appear as localized regions of saturated light (see Figure 1a). These LED reflections are masked out so that they do not interfere with localization of the iris boundaries. First, a mask is created by thresholding on the maximum possible pixel intensity. Second, a morphological open operation is applied to expand the mask and connect neighbouring LED reflections. As can be seen in the upper portion of Figure 1a, large areas of the skin are also sometimes saturated with light. To prevent these areas from being misidentified as LED reflections, a final filtering operation is applied that only retains a masked region (also known as blob or connected component) if its area does lie within a specific range. In the implementation, each connected region in the mask was required to be between 20 and 3,000 square pixels. This range was selected as a best-guess based on empirical observation and may not be optimal. The same is true for the other hard-coded scalar and kernel values presented in this section.

2.1.2. Eyelash Mitigation

Eyelashes can extend over the iris and interfere with localization of the iris boundaries. A simple way to mitigate their impact on iris boundary localization is to apply an open operation with a horizontal (1×8) kernel. This effectively removes or thins dark vertical lines (see Figure 1c for an example).

2.1.3. Contrast Enhancement

Iris images sometimes have low overall contrast, possibly due to being acquired in low-light conditions. To compensate, the contrast is linearly stretched to span the range 0 to 255 at 8-bit depth. This step is performed after the removal of specular highlights. Otherwise, the maximum pixel intensity prior to stretching would always be 255.

2.1.4. Gaussian Blur

The final image preprocessing step is to blur the image to reduce the impact of pixel noise. Blurring is particularly useful at reducing deviations in the computed gradient magnitudes and directions that are used to localize the iris boundaries.

2.1.5. Gradient Information

Both the gradient magnitudes and directions at each pixel are used to localize the iris boundaries. Given an image I presumed to have already been subjected to the above operations, the partial derivatives are denoted $I_x = \frac{\partial I}{\partial x}$ and $I_y = \frac{\partial I}{\partial y}$ and are computed by convolving the 7×7 Sobel kernel [10] with the image. The gradient magnitude and direction at coordinate (x, y) are estimated in the standard way:

$$\left\| \nabla I(x,y) \right\| = \sqrt{I_x(x,y)^2 + I_y(x,y)^2}, \tag{1}$$

$$\theta(x,y) = \operatorname{arctan}_2 \left(I_y(x,y),\; I_x(x,y) \right) \tag{2}$$

2.2. Pupil Localization

The pupil boundary is localized before the limbus. First, all candidate boundary points are identified. A pixel is considered a possible pupil boundary point if it fulfills two criteria: 1) the pixel intensity is below a predefine threshold, and 2) the gradient magnitude is above a predefined threshold. These criteria ensure the point is both dark and forms a strong edge. A binary image is formed by setting all of the pixels that fulfill both of these criteria to one and all other pixels to zero (see the example in Figure 1d). This image is then skeletonized using the method proposed by Zhan-Suen [11] (see Figure 1e).

A standard hough transform [12] is utilized to identify approximately circular shapes in the skeletonized image. Let $A(c_x, c_y, r)$ be the accumulator in the 3-dimensional transform space representing the circle with center (c_x, c_y) and radius r. Given a potential boundary point (x, y) with radius r, the corresponding pupil center is at

$$c_x = x - r \cdot \cos\theta(x, y), \quad (3)$$

$$c_y = y - r \cdot \sin\theta(x, y). \quad (4)$$

Thus, for the given candidate boundary point the accumulator point $A(c_x, c_y, r)$ in the transform space should be incremented. The pupil localization algorithm iterates over all candidate boundary points and possible radii and applies this simple voting strategy. The circle that receives the most votes is then selected to represent the approximate location of the pupil boundary.

The pseudo code is shown below.

Algorithm 1 Pupil localization procedure

```
for all candidate boundary points (x, y) ∈ I do
    for r = 1 to r_max do                         ▷ Iterate up to the max possible pupil radius
        c_x ← x − r · cos θ(x, y)                 ▷ Move along direction of gradient
        c_y ← y − r · sin θ(x, y)

        if I(c_x, c_y) is a candidate boundary point then
            break                                 ▷ Break if hit another potential boundary pixel
        else if r ≥ r_min then
            A(x, y, r) ← A(x, y, r) + 1           ▷ Increment accumulator bin
        end if
    end for
end for
```

The nested for loops iterate over all candidate boundary points and possible pupil radii. The accumulator is incremented each time a pupil center point and radius are deemed valid. The actual implementation applies a simple neighbourhood weighting function around each hough bin $A(c_x, c_y, r)$ when incrementation occurs. The conditional check breaks from the inner loop if another boundary point is met. This mitigates the impact of eyebrows and eyelashes, which often introduce false boundary points. Figure 1f plots the values in A for the example image, though it was necessary to sum over all valid radii at each point (x, y) to make the 3D hough space presentable as a 2D image.

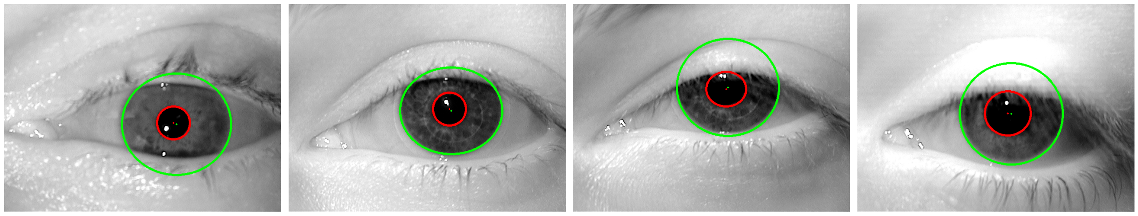


Fig. 2. Examples from ND-0405 of correctly located iris boundaries.

Once an initial circular boundary estimation is identified, the fit is fine-tuned by maximizing the objective function

$$\max_{(a,b,x_0,y_0)} \oint_{a,b,x_0,y_0} \nabla I(x,y) \cdot (\cos\alpha, \sin\alpha)\, \partial s \cdot \left(\frac{\min(a,b)}{\max(a,b)}\right)^{\lambda}. \tag{5}$$

where ∂s follows the path of the ellipse defined by (a,b,x_0,y_0) and α is the angle perpendicular to the ellipse at (x,y). The first term is a generalization of Daugman's integrodifferential operator [13] that allows for elliptical paths with semiaxes a and b. As noted by Daugman, the operator is maximized at circular edges. The second term imposes a penalty proportional to the ellipse's eccentricity. Since pupils are typically nearly circular, a greater penalty is imposed on ellipses that are more eccentric. The exponent λ determines how much weight to apply to the shape of the ellipse and is set to 0.7 in the source code. Rather than solving Equation 5 through brute force, the optimal parameters are found by using the heuristic Nelder-Mead method of convergence [14].

2.3. Limbus Localization

The limbus is assumed to be very nearly concentric with the pupil. First, Equation 5 is maximized via brute force under the constraints that the limbus and pupil boundaries are concentric and the eccentricity is zero (*i.e.*, $a = b$). This leaves the limbus radius as the only free parameter, though its radius is assumed to be no less than 36 pixels greater than the pupil radius. An upper limit of 200 pixels is also imposed. Once the initial parameter values are set, the Nelder-Mead algorithm is again applied (using Equation 5) to optimized the fit, this time without the aforementioned constraints on the location and eccentricity of the limbus. This process could probably be improved slightly by exploiting the fact that the pupil center is tends to be shifted slightly down and toward the nose relative to the limbus center [15, 16]

3. Experimental Results

Accuracy was tested over two datasets, the freely available Notre Dame 0405 dataset [17] and the sequestered Operational Dataset (OPS) 4 used for principle testing in the ongoing Iris Exchange (IREX) 10 evaluation [18]. Although no fine-tuning of the algorithm was

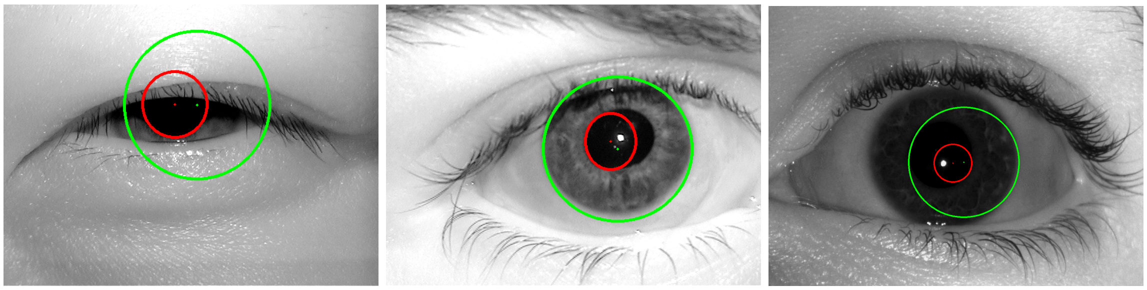


Fig. 3. Examples of improperly located iris boundaries. The left two hail from ND-0405 while the right was acquired in-house to demonstrate the difficulty with localizing the iris boundaries when the iris is very dark.

performed over the datasets, most of the parameter values (e.g. the assumed minimum thickness of the iris) were chosen based on the authors' experience working with the OPS 4 dataset. The segmentation algorithm always assumed there was an iris in the image. For the Notre Dame dataset, the algorithm was able to correctly localize the iris boundaries in 831 out of 837 images (99.3%) based on visual inspection. There were no 'border cases', although the precise locations of the pupil boundaries can be subjective so a few pixels of leeway were permitted. Of the 6 missed cases, the limbus boundary was properly located in 5 while the estimated pupil boundary only partially covered the actually pupil boundary due to its irregular shape. Adjusting the initial step size before applying the iterative Nelder-Mead algorithm might correct this problem. For the OPS 4 dataset, the algorithm was able to correctly localize the iris boundaries in $1,983$ out of a random sampling of $2,000$ iris images (99.2%). A breakdown of the incorrect localizations for this dataset is presented below.

- In 5 cases, part of the limbus was incorrectly identified as part of the pupil boundary.
- In 5 cases, vertical eyelashes extend over the sclera and got misidentified as part of the limbus.
- In 3 cases, the eyelashes obscure the pupil, making it too difficult to localize.
- In 2 cases, the pupil shape is highly irregular, undermining the algorithm's assumption that it is approximately circular.
- In 1 case, a blob of mascara is falsely identified as the pupil.
- In 1 case, the LED reflection inside the pupil is cloudy and diffuse (possibly due to damage to the eyeglasses). This noise introduces false boundaries that interfered with the pupil localization algorithm.

For the 5 cases mentioned in the first bullet, the iris was essentially so dark that the algorithm failed to distinguish it from the pupil (see Figure 3 for an example). Adjusting the pixel intensity threshold referenced in Section 2.2 or improving how contrast enhancement is performed might solve this problem.

4. Summary

An approach to localizing iris boundaries in iris images was presented. Several morphological operations were performed to enhance the salience of the boundaries while reducing the impact of noise. The algorithm was able to correctly localize the iris over 99% of the time for two disparate iris datasets, including the sequestered OPS 4 dataset that is used for performance testing in the IREX 10 ongoing evaluation. The source code is free for developers and universities to download and use.

5. Notes

The full source code is available on GitHub with no license requirements at https://github.com/gwquinn/IrisFinder.

References

[1] Iris Recognition: Impact and Use Policy, https://www1.nyc.gov/assets/nypd/downloads/pdf/public_information/post-final/iris-recognition-nypd-impact-and-use-policy_4.9.21_final.pdf. Accessed: 3/6/24.

[2] Government of India Unique identification authority india, https://uidai.gov.in/. Accessed: 3/6/24 Available at https://uidai.gov.in/.

[3] Daugman J (2007) New methods in iris recognition. *IEEE Trans Systems, Man, Cybernetics B* 37(5):1167–1175.

[4] The Eyes Have It: Iris Biometric Added to Next Generation Identification System, https://www.fbi.gov/news/stories/fbi-adds-iris-biometric-to-next-generation-identification-system-121120. Accessed: 3/6/24.

[5] Alaslani M, Elrefaei L (2018) Convolutional neural network based feature extraction for iris recognition. *International Journal of Computer Science and Information Technology* 10:65–78. https://doi.org/10.5121/ijcsit.2018.10206

[6] Minaee S, Abdolrashidiy A, Wang Y (2016) An experimental study of deep convolutional features for iris recognition. *2016 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, pp 1–6. https://doi.org/10.1109/SPMB.2016.7846859

[7] Bendale A, Nigam A, Prakash S, Gupta P (2012) Iris segmentation using an improved hough transform. *Eighth International Conference on Intelligent Computing (ICIC)*. https://doi.org/10.1007/978-3-642-31837-5_59

[8] Statistics on the Location and Size of Iris Boundaries in the OPS4 Iris Dataset, https://www.nist.gov/system/files/documents/2021/11/21/ops4.pdf. Accessed: 3/6/24.

[9] Bradski G (2000) The OpenCV Library. *Dr Dobb's Journal of Software Tools* .

[10] Pratt WK (2007) *Digital image processing: PIKS Scientific inside*. Vol. 4 (Wiley Online Library).

[11] Zhang TY, Suen CY (1984) A fast parallel algorithm for thinning digital patterns.

Commun ACM 27(3):236–239. Available at http://dblp.uni-trier.de/db/journals/cacm/cacm27.html#ZhangS84.

[12] Duda RO, Hart PE (1972) Use of the hough transformation to detect lines and curves in pictures. *Commun ACM* 15(1):11–15. Available at http://dblp.uni-trier.de/db/journals/cacm/cacm15.html#DudaH72.

[13] Daugman J (2003) The importance of being random: statistical principles of iris recognition. *Pattern Recognition* 36(2):279–291. Available at http://dblp.uni-trier.de/db/journals/pr/pr36.html#Daugman03.

[14] Olsson DM, Nelson LS (1975) The nelder-mead simplex procedure for function minimization. *Technometrics* 17(1):45–51. Available at https://amstat.tandfonline.com/doi/abs/10.1080/00401706.1975.10489269.

[15] Grother P, Tabassi E, Quinn G, Salamon W (2009) Performance of iris recognition algorithms on standard images. *NIST Interagency Report* 7629(1-120):33–34.

[16] Czajka A, Bowyer KW, Krumdick M, VidalMata RG (2017) Recognition of image-orientation-based iris spoofing. *IEEE Transactions on information Forensics and Security* 12(9):2184–2196.

[17] Bowyer KW, Flynn PJ (2016) The nd-iris-0405 iris image dataset, https://cvrl.nd.edu/projects/data. 1606.04853.

[18] IREX 10: Identification Track, https://pages.nist.gov/IREX10. Accessed: 3/6/24.