Iris Detection and Normalization

Thesis · January 2011

1 author:

Todor Kazakov

1 PUBLICATION **6** CITATIONS

SEE PROFILE

IRIS DETECTION AND NORMALIZATION

by

TODOR KAZAKOV

A thesis submitted to
University of Birmingham
for the degree of
BENG COMPUTER SCIENCE/SOFTWARE ENGINEERING

School of Computer Science
University of Birmingham
May 2011

Abstract

Biometric systems use the unique body and behavioural features of individuals for identity recognition and verification. Iris recognition systems particularly exploit the fact, that humans have unique patterns in their irises. For these to be recognizable by pattern matching algorithms, the iris tissue in images has to be isolated and transformed to a normalized form. In this report the detection and normalization problems are explored and time efficient solution algorithms are developed. The iris boundary detection algorithm is a hybrid approach of John Daugman's and Mahboubeh Shamsi's algorithms and uses their best features, providing efficient and accurate boundary detection. The initial objective of robustness and time efficiency was fulfilled, and the algorithm achieves $\approx 96\%$ correct pixel detection with an average detection time of $\approx 0,7$ seconds. The detected region of interest - the isolated iris tissue, is unwrapped - converted from Cartesian to Polar coordinates, using a newly derived formula, based on Daugman's Rubber-Sheet model.

CONTENTS

CHAPTER 1

INTRODUCTION

Iris recognition systems are used for biometric identification. Such systems need to acquire iris images, isolate the iris tissue in those, and apply pattern matching algorithms. The scope of this project covers the iris detection (segmentation) and the normalization of the found region of interest. Because iris image acquisition is not a simple process and requires additional hardware, CASIA Iris image databases [6] V1.0 and V3.0 were used for the development of the segmentation and normalization algorithms.

1.1 Biometrics

The word biometrics is derived from ancient Greek bio meaning life and metric meaning to measure. Biometrics is the study of unchangeable measurable biological features [1], used for verification and identification of individuals. The set of biological features are divided into two main subsets - physiological data and behavioural features. The physiological data is characterized by a measure describing persons facial appearance, eye's colour, iris patterns, fingerprints, vein structure, the pitch of a voice, body odour or DNA. Behavioural biometrics explores behavioural patterns, identified in writing style, typing speed, walking rhythm and others. There is a lot of ongoing research in the development and improvement of these technologies. Biometrics is strongly related to recognition of people's identity. Because of the high reliability of certain biometric systems, they are

employed in airports, police departments and highly secured facilities to fight crime, terrorism and illegal immigration [9]. Each method for identification has its advantages and drawbacks - the main reason behind the existence of biometric applications, which incorporate more than one. This project explores the important process of iris segmentation - the process of isolation of the iris tissue in eye images, on which pattern recognition algorithms are applied.

1.2 Eye Anatomy

To be able to get into more depth in how iris recognition works, why iris recognition is even possible and why it is important in section 1.4, human's eye anatomy has to be briefly explained. The pupil, iris and sclera are the most visible eye structures, which are seen when somebody looks at us (Figure. 1.1).

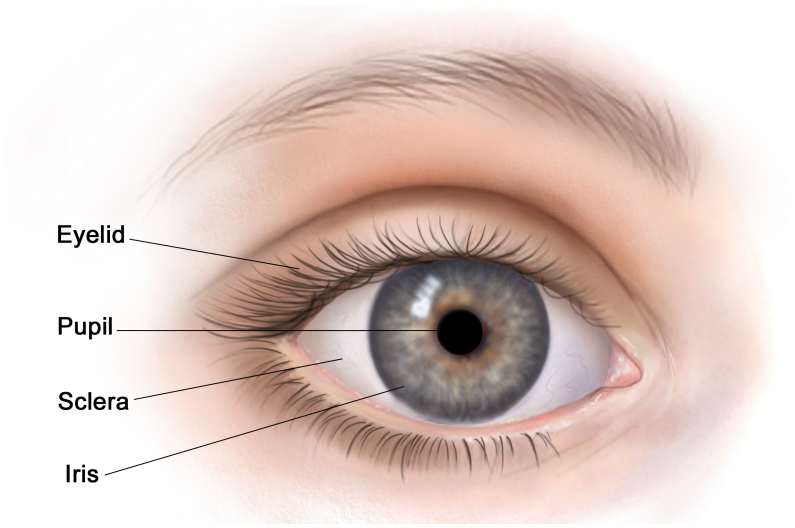


Figure 1.1: Human Eye

The iris is the coloured part of the eye - it is a circular (or elliptical) muscle, behind the cornea. It controls light levels inside the eye similar to the aperture on a camera [2]. The round (circular or elliptical) opening in the centre of the iris is called the pupil. The iris is embedded with tiny muscles that dilate (widen) and constrict (narrow) the pupil

size. The colour, texture and patterns of the iris are unique, thus providing excellent method for identification with a false match rate as low as 1 in 10^{15} comparisons [5]. The iris patterns are formed randomly during embryonic gestation, therefore even genetically identical twins have different iris patterns [16]. The sphincter muscle lies around the very edge of the pupil. In bright light, the sphincter contracts, causing the pupil to constrict. The dilator muscle runs radially through the iris, like spokes on a wheel. This muscle dilates the eye in dim lighting [2]. The eyelids protect the eyes from the environment, injury and light. Upper and the lower eyelid may occlude significant parts of the iris. There are eyelashes on the edge of the eyelid , which protect the eye from debris and are sensitive to touch, providing an early warning when an object is near to the eye. Eyelashes and eyelids may result in false iris detection and/or false rejection in iris recognition.

1.3 What is Iris Recognition

Iris Recognition is the process of biometric identification of a person, by recognition of iris patterns. It is a relatively new technology, patented in 1994 by Prof. John Daugman [3]. As explained in section 1.2, iris patterns are formed during embryonic gestation and are stable during one's life - do not drastically change over time as for example faces change. The stability and uniqueness of the irises make iris recognition probably the most reliable technology for identification. Daugman et al. notes in [4], that the relation between inter-class and intra-class variability makes iris recognition possible: objects can be reliably classified, because the variability among different instances of a given class is less than the variability between different classes. The variation of iris patterns is enormous. Even though the size of the iris changes in different lighting conditions, non-affine transformations of the iris patterns caused by pupil dilation are easily reversible, because the iris is an elastic tissue.

1.4 Iris Recognition Process Explained

Most of today's iris recognition systems are based on the principles described in the early iris recognition patents by John Daugman.

The process begins with image acquisition. An image of the subject's iris is usually taken at Near Infrared (NIR) illumination, discarding the colour. Although the colour in the iris, caused by the pigment melanin, is a great source of information for iris recognition, visible light image acquisition is often avoided, because reflections in the cornea pose some difficulties for the process [12].

The acquired image is then pre-processed - it might undergo a focal analysis, and if the subject is out of focus the recognition process starts from the beginning. At this stage the image might also be enhanced - by histogram equalization and/or brightness and contrast adjustment.

Segmentation - there are various approaches of how to segment the digital image of an eye. The main task is to isolate the iris region. This is done by detecting two circles - one for the limbic boundary (between iris and sclera) and another, inferior to the first one, for the pupillary boundary (between pupil and iris). The centres and radii of these circles must be found. This is done by using circle detection algorithms. This process is possible, because the high contrast between pupil, iris and sclera. Normally eyelids occlude irises and eyelashes and specular reflections are present in some images - due to these factors, problems might be present in ensuing stages of detection algorithms. To ensure the accuracy of pattern matching algorithms, it is a good practice to detect eyelids at this stage and to add eyelashes and specular reflections to a noise map. Once the region of interest is established, the iris tissue will be separated from all other parts of the eye - pupil, sclera, eyelids and eyelashes, leaving only the usable information for further processing.

During the normalization step, the isolated region of interest is converted to a form with a predefined size (specified by angular and radial resolution), using Cartesian to polar transform.

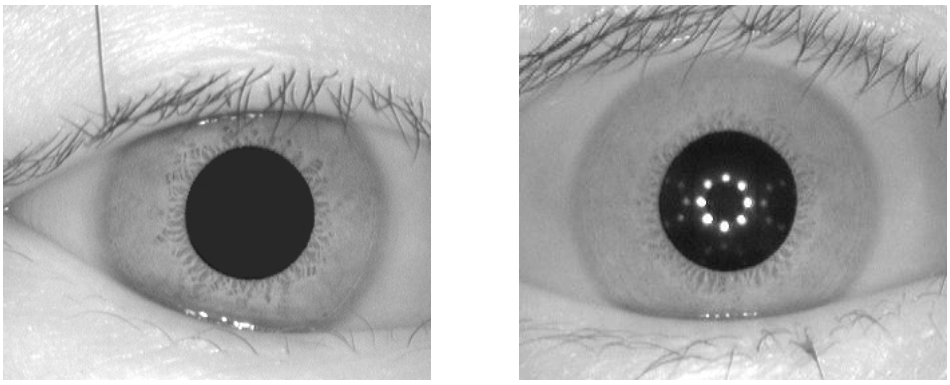
At the encoding stage, the normalized image is encoded into a binary template, called iris code. This is usually done by phase-quadrant demodulation of the signal of the normalized image convolved with wavelets.

During the matching stage, the generated binary template is compared to others, stored in the database to identify the subject. The Hamming distance between iris templates is calculated, where a Hamming distance below some value, means a positive identification of the subject.

CHAPTER 2

RESOURCES

The iris images used for testing and evaluation of the developed algorithms are part of CASIA Iris Image Databases V1.0 and V3.0 (Interval). Due to the complexity and variations in lighting conditions in iris image acquisition, the iris detection algorithm is guaranteed to work without modifications only on the selected iris image databases.



(a) CASIA V1.0 (b) CASIA V3.0

Figure 2.1: CASIA Iris Image Database Sample Images

2.1 CASIA V1.0

CASIA Iris Image Database Version 1.0 (CASIA-IrisV1) includes 756 iris images from 108 eyes. For each eye, 7 images are captured in two sessions with our self-developed device CASIA close-up iris camera. All images are stored as BMP format with resolution 320 x 280.

In order to protect our IPR in the design of our iris camera (especially the NIR illumination scheme), the pupil regions of all iris images in CASIA-IrisV1

were automatically detected and replaced with a circular region of constant intensity to mask out the specular reflections from the NIR illuminators (see Figure 2.1(a)). Such editing clearly makes iris boundary detection much easier but has minimal or no effects on other components of an iris recognition system, such as feature extraction and classifier design.

2.2 CASIA V3.0

CASIA-IrisV3 includes three subsets which are labelled as CASIA-Iris-Interval, CASIA-Iris-Lamp, CASIA-Iris-Twins. CASIA-IrisV3 contains a total of 22,034 iris images from more than 700 subjects. All iris images are 8 bit gray-level JPEG files, collected under near infrared illumination.

Iris images of CASIA-Iris-Interval were captured with our self-developed close-up iris camera. The most compelling feature of our iris camera is that we have designed a circular NIR LED array, with suitable luminous flux for iris imaging. Because of this novel design, our iris camera can capture very clear iris images (see Figure 2.1(b)). CASIA-Iris-Interval is well-suited to study the detailed texture features of iris images.

CHAPTER 3

BACKGROUND

There is a lot of ongoing research on methods for iris recognition, important part of which is the iris segmentation process. During it, the iris tissue is isolated - the boundaries between the pupil and the iris as well as this between iris and sclera are detected. Often, problems for recognition systems are introduced by eyelashes and eyelids, which occlude the eye and introduce noise in the normalized iris region (see section 5.4), resulting in a false rejection. Although, eyelashes have to be extracted and eyelids have to be detected for a successful real-world application, these are outside of the scope of the project.

3.1 Iris Detection

In the general case, the pupil and iris have circular form with some exceptions, where they might be elliptical. To isolate the iris tissue, the centres and radii of pupil and iris have to be detected in the iris image (Figure 2.1(a)) - a challenging problem, but a necessary step before pattern recognition. To find these parameters, some kind of circle detection algorithm should be employed. Few approaches were reviewed and their advantages were combined to form the final product.

3.1.1 Daugman's Integro-Differential Operator

As described in section 1.4, iris detection relies on the fact, that the contrast between pupil and iris, and iris and sclera is high. John Daugman, the pioneer in iris recognition, proposed [4] an elegant and effective solution, to detect the circular boundaries' parameters - the circular integro-differential operator:

$$\max_{(r,x_0,y_0)} \left| G_\sigma(r) * \frac{\partial}{\partial r} \oint_{r,x_0,y_0} \frac{I(x,y)}{2\pi r} ds \right| \tag{3.1}$$

The explanation given by Daugman et al. [4] is:

> where I(x,y) is an image such as containing an eye. The operator searches over the image domain (x, y) for the maximum in the blurred partial derivative with respect to increasing radius r, of the normalized contour integral of $I(x, y)$ along a circular arc ds of radius r and center coordinates (x_0, y_0). The symbol $*$ denotes convolution and $G_\sigma(r)$ is a smoothing function such as a Gaussian of scale σ. The complete operator behaves in effect as a circular edge detector, blurred at a scale set by σ, which searches iteratively for a maximum contour integral derivative with increasing radius at successively finer scales of analysis through the three parameter space of center coordinates and radius (x_0, y_0, r) defining a path of contour integration.

This operator can be modified to search for arcs instead of circles, to detect the upper and lower eyelids.

3.1.2 Shamsi's Difference Function

Shamsi et al. [15] proposed a difference function (Eq. 3.2), which works in a similar way as Daugman's operator. It searches for maximal difference between the inner and outer pixels. A number of points (Circe Sample - CS) over the circular path are chosen and the difference between top, bottom, lef, right and diagonal pixels is calculated (Figure 3.1.2).

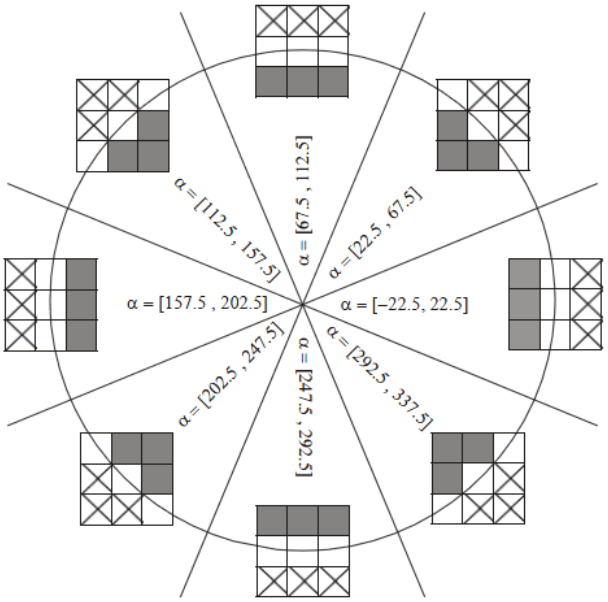


Figure 3.1: Difference between inner and outer pixels

$$\max_{(x_c,y_c,r)} \sum_{j=1}^{CS} diff(x_j, y_j) \tag{3.2}$$

for all potential x_c, y_c and r, where

$x_j = x_c + r \times \cos(\alpha_j), y_j = y_c + r \times \sin(\alpha_j), \theta_j = \left(\left\lfloor \frac{\alpha_j + 22.5}{45} \right\rfloor \times 45 \right)$

$diff(x_j, y_j) = \left| \sum_{\tilde{i}=1}^{3} \sum_{\tilde{j}=1}^{3} \tilde{I}_{\tilde{i},\tilde{j}} \right|$

$$\tilde{I} = \begin{bmatrix} \sin((\theta_j - 45) \times (\frac{2\pi}{360})) & \sin((\theta_j) \times (\frac{2\pi}{360})) & \sin((\theta_j + 45) \times (\frac{2\pi}{360})) \\ \sin((\theta_j - 90) \times (\frac{2\pi}{360})) & 0 & \sin((\theta_j + 90) \times (\frac{2\pi}{360})) \\ \sin((\theta_j - 135) \times (\frac{2\pi}{360})) & \sin((\theta_j) \times (\frac{2\pi}{360})) & \sin((\theta_j + 135) \times (\frac{2\pi}{360})) \end{bmatrix} \circ$$

$$\begin{bmatrix} I(x_j - 1, y_j - 1) & x_j, y_j - 1 & x_j + 1, y_j - 1 \\ I(x_j - 1, y_j) & x_j, y_j & x_j + 1, y_j \\ I(x_j - 1, y_j + 1) & x_j, y_j + 1 & x_j + 1, y_j + 1 \end{bmatrix}$$

3.1.3 Circular Hough transform

An automatic segmentation algorithm based on the circular Hough transform is employed by Masek et al. [10]:

The Hough transform is a standard computer vision algorithm that can be used to determine the parameters of simple geometric objects, such as lines and circles, present in an image. The circular Hough transform can be employed to deduce the radius and centre coordinates of the pupil and iris regions. Firstly, an edge map is generated by calculating the first derivatives of intensity values in an eye image and then thresholding the result. From the edge map, votes are cast in Hough space for the parameters of circles passing through each edge point. These parameters are the centre coordinates x_c and y_c, and the radius r, which are able to define any circle according to the equation.

A maximum point in the Hough space will correspond to the radius and centre coordinates of the circle best defined by the edge points.

3.1.4 Using binarization and morphology operators

Circle detection algorithms, based on morphology operators can be a very time efficient detection method. These are employed by Mohammed et al. [11], Huan et al. [8], Rajpathak et al. [13].To apply these operators, first the original image must be binarized by picking an appropriate threshold value. Pixels with values greater than certain level are set to white and ones with a lower values to black (Figure 3.2). A series of morphology operators are then applied to the binarized image to detect the pupillary and limbic boundaries. Morphological approach might not be the perfect solution for real-world applications, because picking the right threshold value can be a rather difficult task. In such applications, the lighting conditions may vary, posing serious risk for false detection of pupil and iris. Typically, complex algorithms for histogram analysis are being executed on each images to automatically establish a threshold value. This may result in unwanted delays in the segmentation process. If the segmentation is applied on certain database only this value can be established by manually analysing histograms of a training set.

Figure 3.2: Black and white image of an iris

3.2 Average Square Shrinking

In this approach, the iris image is shrunken by a certain shrinking factor (S_f) until it reaches a specified size. Each shrunken image is stored in a data structure for further processing. The circle detection algorithm starts with the smallest image. Because the pupil is the darkest area in the original image, when the image is scaled to a size of 10 x 10 pixels and smoothed by Gaussian blur, the darkest pixel near the centre of the in this image is considered to be the centre of the pupil. The radius of the pupil in image of this size is estimated to be in the range from 1 to 2 pixels [15] - i.e. only two circles will be analysed in the smallest image. The results of the detection are the three parameters of the circle for pupillary boundary. These are then converted (multiplied by the S_f) and are set as reference points for pupil and iris detection on higher shrunk images, where circle detection algorithm is applied again. This approach is proven to be very efficient, because the limits for the circles' parameters are tightly bond with the value of $S_f = 3$ (Shamsi et al. [15]). This approach is a search space reduction strategy combined with a circle detector 3.1.2.

CHAPTER 4

PROJECT REQUIREMENTS

The following project requirements were set at the beginning of the project, identifying the main goals. They are divided into two categories - functional and non-functional requirements, specifying software products's required functionality, efficiency and limitations.

4.1 Functional requirements

F. 1 Iris Segmentation (Iris boundary detection)

F. 1.1 The algorithm must be able to detect iris boundaries

F. 1.1.1 The algorithm must return radius and coordinates of the centre of a circle for the boundary between pupil and iris

F. 1.1.2 The algorithm must return radius and coordinates of the centre of a circle for the boundary between iris and sclera

F. 1.2 The algorithm must return the execution time of the segmentation process

F. 1.3 The algorithm should shrink the original image of the iris by a certain shrinking factor until it reaches predefined size and store the resulting images in a data structure for further processing

F. 1.3.1 Each image is processed in a certain stage

F. 1.3.2 Default boundary search space for iris and pupil should be established. The search space is represented by ranges of radii and potential centre points of circles.

F. 1.3.3 The default boundary search space for every other image should be calculated relative to the previous stage

F. 1.3.4 Users should be able to modify the default calculated search space for each boundary

F. 2 The system should be able to unwrap the detected iris tissue according to Daugman's Rubber-Sheet model.

F. 3 There must be a graphical user interface, providing following features:

F. 3.1 An image should be able to be loaded and displayed

F. 3.2 The iris segmentation algorithm could be performed on the loaded image

F. 3.3 The normalization algorithm could be performed on the loaded image, using detected the parameters of the circles of the iris boundaries.

F. 3.4 The default search space for each stage of detection should be modifiable through the GUI

4.2 Non-functional requirements

NF. 1 The iris segmentation algorithm should achieve approximately 90% accuracy, measured by Jaccard's similarity coefficient, between the sets of pixels in detected and ground truth boundaries.

NF. 2 The average segmentation time should be less than 1 second.

NF. 3 The average normalization time should be less than 0.1 seconds.

NF. 4 The algorithms should be implemented in MATLAB

NF. 5 The algorithm must be easily modifiable

NF. 6 The segmentation and normalization algorithms, should be in a modular structure, providing easy incorporation in other applications

CHAPTER 5

PROBLEM ANALYSIS

5.1 Detecting the circles in the image

To detect the circles in the iris images, any of the reviewed in section 3.1 approaches can be used. Some of the algorithms work better than others. An appropriate one has to be chosen, to satisfy the project's requirements, . The developed solution is necessary to be easily modifiable to work on any iris image database.

Because usually in most iris image databases, there is a high contrast between pupil, iris and sclera, both approaches - the one employed by Shamsi et al. [15] and the one proposed by Daugman et al. [4] would be appropriate to detect irises boundaries.

The described in section 3.1.1 integro-differential operator, employed in Daugman's algorithms, was proven to be very efficient, achieving excellent detection results, with more than 95% of successful pupil and iris detection [15]. The operator searches for the circular path where there is maximum change in pixel values, by varying the radius r, centre x and y position of the circular contour.

Optimizations were necessary to combine the accuracy with appropriate time efficiency results. The great accuracy, the ability to work on digital eye images from any database without significant customization, and the optimization possibilities available were the main factors, which led to the decision to use this method as a basis of the developed segmentation algorithm.

5.2 Search space reduction

It is impossible or at least a time consuming process to detect all possible circles in an image. As mentioned in section 1.2, pupillary and limbic boundaries are circles. Circles are mathematically represented by three parameters - position of the centre along the x-axis, position of the centre along the y-axis and radius of the circle. These parameters have to be computed with a circle detection algorithm. To present a time efficient solution, the image space on which the algorithm will work has to be reduced as possible. It will be referred as a search space. The search space is defined by potential centre points and radius range. To search for the best circle within all the circles in an image of size 320 x 280 pixels, whose centres are in the image, the search space for the three parameters of the circle (x, y, r) will have the following boundaries:

$$x \in [1, 320]\,; y \in [1, 280]\,; r \in [1, 320]\,;$$

The algorithm will run for each x, for each y, for each r, having a computational complexity of $O(n^3)$, and if Daugman's Integro-differential Operator is employed for circle detection, the average of value of all the pixels, lying on the circumference of the circle, for all $320 \times 280 \times 320 = 28672 \times 10^3$ circles has to be computed.

To optimize the algorithm, boundaries for the three parameters are set. This is possible, because:

- In databases of iris images, such as CASIA, the minimum and maximum size of the irises can be estimated by observing a training set.
- The centres of pupil and iris in such databases are near to the centre of the image.
- There is a relation between the size of iris and pupil (see section 6.5)

Selecting the 11 by 11 neighbourhood around the centre of the image for potential centres is effective for CASIA iris image database. The minimum and maximum sizes of the irises for the same database are estimated to be 80 and 150 (Masek et al. [10]),

respectively. Although, the algorithm is still working within the same computational complexity class, the number of circles analysed just for the detection of the limbic boundary is $11 \times 11 \times 70 = 8470$. In the prototypes of the algorithm, this approach has been adopted. Even though the execution was relatively fast ($\approx$ 10 seconds), further optimizations are necessary in order to meet project's non-functional requirements.

Employed in this project, and proven to be more effective approach for search space reduction, the "Average Square Shrinking" approach has been proposed by Shamsi et al. [15] (described in section 3.2). It is used to reduce the search space as much as possible.

To balance between efficiency and accuracy, in this project, the parameters of the shrinking algorithm are modified (size of the neighbourhood around the reference points and ranges of radii in each of the shrunken images). The average number of circles analysed for the detection of both pupillary and limbic boundary is $\approx$ 1945 (see section 9.4). The most significant disadvantage of adopting this approach is that incorrect boundary detection at an early stage will almost inevitably cause incorrect pupil/iris detection at later stages.

5.3 Pixels' coordinates calculation

In the proposed method for circle detection, employed by Daugman et al. [4], the integro-differential operator's (Eq. 3.1) computation is achieved through calculation of the average value of the pixels, which lie on the circumference of every analysed circle. To do this, their location in the image has to be calculated.

There are various algorithms to calculate the coordinates of the points which lie on the circumference of a circle. The simplest approach is to use trigonometry. For every θ in the interval $[0 : 2\pi]$, the coordinates can be computed using the following formula:

$$X = O_x + r \times \cos(\theta) \tag{5.1}$$

$$Y = O_y + r \times \sin(\theta) \tag{5.2}$$

The disadvantage of this approach is that it is slow - trigonometric functions are computationally expensive. The speed for today's computers is not an issue, if only one circle is to be drawn, but this is not the case. An optimized solution, to this problem, which uses integer algebra only is described in section 6.3.

5.4 Normalization - Daugman's Rubber Sheet Model

Once the limbic and pupillary boundaries are detected, during the normalization, the iris is converted to a standard (predefined, fixed) size, and then encoded in an iris template for matching between the iris templates. Typically, the process is a conversion from Cartesian to non-concentric polar representation. The polar coordinate grid is not always concentric, because in most eyes pupil is not central to the iris. Daugman et al. [4] derived the formula:

$$I(x(r,\theta), y(r,\theta)) \rightarrow I(r,\theta) \tag{5.3}$$

With:

$$x(r,\theta) = (1-r)x_p(\theta) + rx_l(\theta) \tag{5.4}$$

$$y(r,\theta) = (1-r)y_p(\theta) + ry_l(\theta) \tag{5.5}$$

The coordinates of the pupil and iris boundaries along the θ direction are x_p, y_p, x_l, y_l.

In different eye images from the same subject, changes in the iris pattern occur due to pupil dilation or constriction. The conversion to a dimensionless form also ensures the iris pattern is not influenced by magnification and image distance acquisition factors. John Daugman's rubber sheet model ensures the proper handling of the matter due to the specifics of the iris. It accounts size inconsistencies and pupil dilation, but does not compensate for rotational inconsistencies [15]. These can be eliminated at the matching

stage with simple bit-shifting of the encoded iris template.

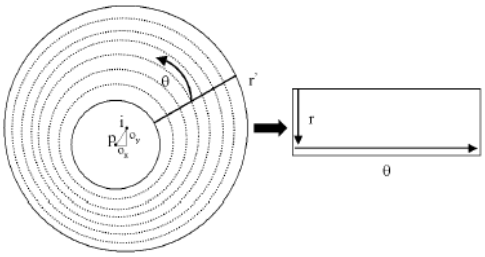
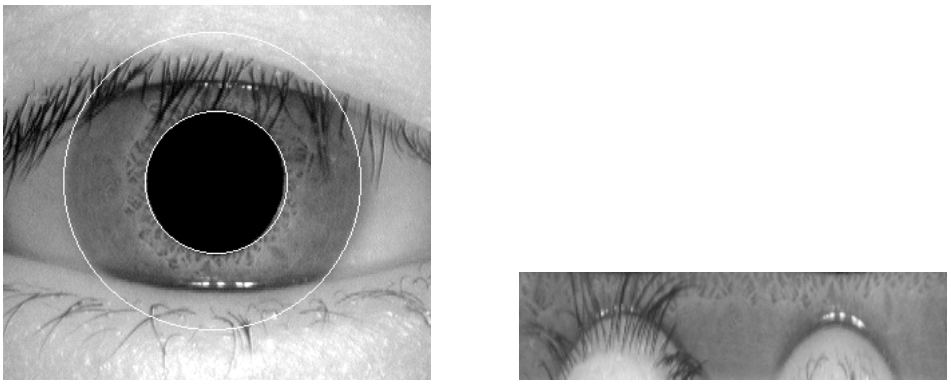


Figure 5.1: Rubber Sheet Unwrapping

Figure 5.2 shows an image of an iris with detected pupillary and iris boundaries and the normalized region. As seen on Figure 5.2(b), eyelid occlusion and eyelash presence in the iris region can cause artefacts in the normalized image.



(a) Original iris image (b) Normalized image

Figure 5.2: Original and normalized iris image

Daugman et al. [4]:

> The points between the inner and outer iris boundaries are interpolated linearly by a homogeneous rubber sheet model, which automatically reverses the iris pattern deformations caused by pupillary dilation or constriction. Under assumptions of uniform iris elasticity (which may be questionable), this normalization maps the iris tissue into a doubly-dimensionless coordinate system. The homogeneous rubber sheet model assigns to each point in the iris, regardless of iris size in the image and of pupillary dilation, a pair of dimensionless real coordinates r,θ where r lies in the unit interval $[0, 1]$ and θ is the angular variable, cyclic over $[0 : 2\pi]$. This homogeneous rubber sheet model maps the iris into a dimensionless, normalized coordinate system that is size-invariant, and therefore invariant to changes in the target distance and the optical magnification factor, as well as invariant to the position of the eye in

the image, and invariant to the pupil dilation (assuming uniform iris elasticity). It is not strictly polar, because it makes no assumption that the pupil and iris are concentric (indeed the pupil's actual centre is usually nasal, and inferior, to the centre of the iris), nor even that their boundaries are circular. An important aspect of this method of normalization is that it does not introduce unnecessary cuts in the intrinsically continuous and cyclic angular variable, which would interrupt subsequent convolution, as occurs in other methods that explicitly unwrap the iris into a rectangular domain.

CHAPTER 6

SOLUTION DESIGN

It was known that Daugman's algorithm achieved great accuracy in terms of iris/pupil detection rate, but to decide on the algorithm, which is suitable to meet the requirements of the project, prototypes of different algorithms were built. During the research stage (see Project Management - chapter 8), it was decided that Daugman's integro-differential operator was suitable to satisfy project's requirements. When early prototypes of the segmentation algorithm were built, it was noticed that optimizations were necessary to achieve detection times specified in project requirements (section 4.2).

6.1 Prototypes

The evaluation of the prototypes in terms of accuracy was subjective - the segmentation was performed on a small training set consisting of 60 images from CASIA V1.0 iris image database (see section 2.1), where the detection was considered to be either correct, or incorrect. In contrast, a quantitative evaluation method was used to evaluate the boundary detection of the final product (section 9.1).

- Prototype number one - Because of the accurate and time efficient results of the algorithm proposed by Shamsi et al. [15], it was considered suitable to meet the requirements of the project. The results of the prototype showed that although the algorithm is very efficient in terms of accurate boundary detection, it is working

slower than expected, when implemented in MATLAB. It was considered, even with the possible optimizations, the average execution time will not be low enough to satisfy the non-functional requirements. The average execution times reached in [15] are considered implementation specific, but implementation details beyond the algorithm diagram are not disclosed. As explained in section 3.1.2, the algorithm uses computationally expensive trigonometry functions to calculate the difference between inner and outer pixels of every processed circle.

- Prototype number two - another prototype was built, following Daugman's algorithm [4]. It showed high segmentation accuracy, performed on the training set, with still relatively high execution times. It was decided that this approach is more suitable to be implemented in the final product, because the code allowed more optimizations (fully described in sections 6.3, 6.4, 6.5).

6.2 Proposed method

The proposed method is based on Daugman's iris segmentation algorithm [4]. The integro-differential operator 3.1 was used as a circle detector in the images. It searches for the maximum difference in the mean value of the pixels, which lie on the contour of the circle in the three parameter search space (x, y, r). To optimize and improve the algorithm, series of modifications were made to the original algorithm - integer algebra only is used to generate the set of points lying on a circumference of a circle, eyelid occlusion errors are minimized, and search space is reduced. The most significant modification was to employ search space reduction strategy, described in section 3.1.2 - the Average Square Shrinking Approach [15]. To fully explain how this was done, the detection process is explained step-by-step in section 6.5.

6.3 Midpoint circle algorithm

To calculate the coordinates of the pixels, which lie on circles outline, using integer algebra only, the midpoint circle algorithm is implemented. It is used primarily for circle drawing, but in this case it used to generate the coordinates of the outline pixels. The midpoint is a standard, very fast computer vision algorithm, which uses integer algebra only and the eight-way symmetry of a circle to generate it. It plots 1/8 part of a circle (for example from 90 to 45 degrees) and then it uses symmetry to mirror the points (see Figure 6.1). To draw the 1/8 part of the circle, unit steps in the positive x direction are taken and decision parameter is used to determine which of the two possible y (y or $y+1$) is closer to the true circle path at each step. Once x and y are calculated, they are mirrored - the 8 points, which lie on the circle are $(x, y), (y, x), (x, -y), (-y, x), (-x, -y), (-y, -x), (-x, y)$ and $(y, -x)$ [7]

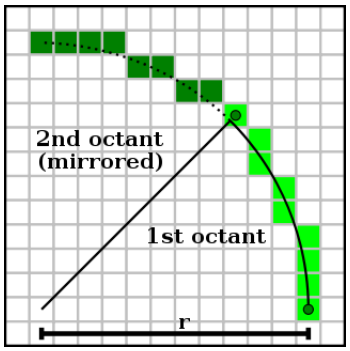
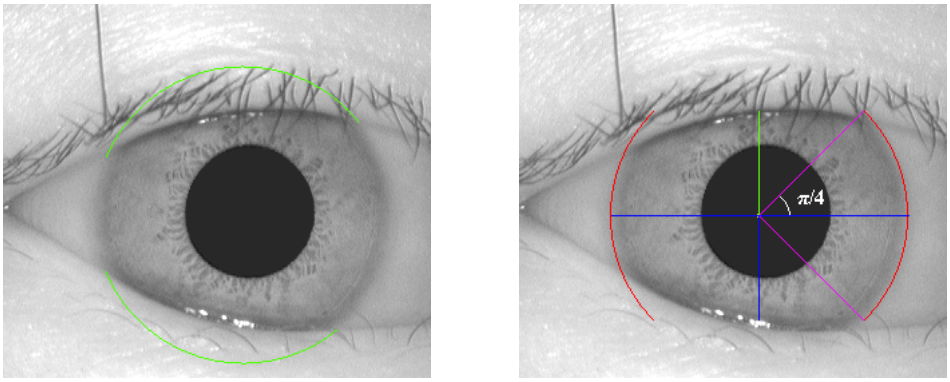


Figure 6.1: Midpoint Circle Algorithm

6.4 Reduction of false detection due to eyelid occlusion

Eyelids occlusion often results in inaccurate iris segmentation. Daugman's Integro-differential operator takes the average values of the pixels, which lie on every processed circle circumference. As shown on Figure 6.2(a), if eyelids occlude part of the iris, the values of pixels, which are on the eyelids, are taken for the computations (the green pixels), where

they should be considered as noise, and therefore, be excluded. To reduce the noise, it is considered to be a good practice to calculate the mean value of the pixels on circumference of the circle, which are only in the interval $[0, \pi/4] \cup [3\pi/4, 5\pi/4] \cup [7\pi/4, 2\pi]$. In the "worst case", when there is no eyelid occlusion present on the image, this method will lead to the exclusion of non-noise pixels. This does not pose problem, because large enough portion of "valid" pixels is still being analysed.



(a) Pixels on the eyelids (b) Pixels used

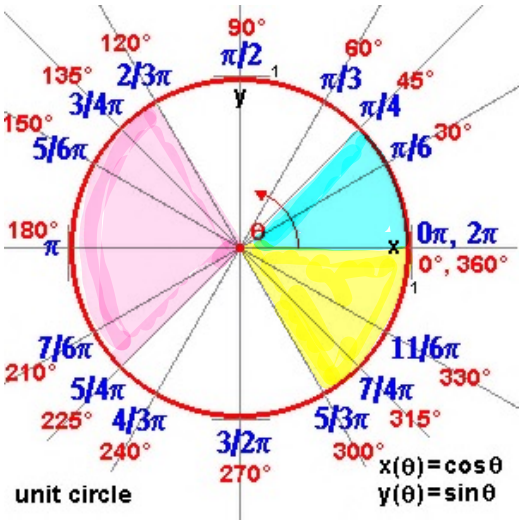
Figure 6.2: Pixels lying on the eyelids and pixels used in the computations

Because the coordinates of the pixels are generated by the midpoint circle algorithm (section 6.3), but not by a function of the angle θ and radius, a method for exclusion of the noise pixels is proposed. It does not rely on the connection between the angle and the position of the pixel in consideration.

For each circle processed, to calculate the integro-differential operator (Eq. 3.1), the following equation for l (the green line on Figure 6.2(b)) is calculated:

$$l = r \times \sin(\pi/4) \tag{6.1}$$

If r is the radius of the circle, O_y is the coordinate of the circle centre along the y-axis and y is the coordinate of the pixel in consideration along the y-axis, all pixels, where $y < Oy - l$ or $y > Oy + l$ are ignored (the red pixels on Figure 6.2(b)).


90°
π/2
120°
2/3π
135°
3/4π
150°
5/6π
180°
π
7/6π
210°
5/4π
225°
4/3π
240°
3/2π
270°
60°
π/3
45°
π/4
30°
π/6
0π, 2π
0°, 360°
11/6π
330°
7/4π
315°
5/3π
300°
y
x
θ
1
1
unit circle
x(θ)=cosθ
y(θ)=sinθ

6.5 Proposed algorithm

As explained in section 5.2, the detection of all circles in an image is impossible and the limits for each parameter should be set to reduce the search space. This will result in a lower number of processed circles, leading to lower execution times. Pixels from a fixed sized neighbourhood (usually 11 by 11) around the centre of the image can be taken as potential points for centre of the pupil and the iris and the ranges of radius for the pupil and the iris can be estimated by analysing the iris image database used. But in the proposed approach, the search space is even further decreased. This is done to reduce the average execution time, by adopting the Average Square Shrinking approach (Shamsi et al. [15]). It is explained in details in section 3.1.2.

The process begins by shrinking the iris image by a certain predefined shrinking factor (S_f), which for this project was chosen to be $S_f = 2$*. The shrinking process is repeated until the size of the image is lower than or equal to 20 x 20 pixels*. Each intermediary image of the shrinking process is saved in a data structure for further processing. Each image corresponds to boundary detection at certain detection stage, where the first stage is the detection in the smallest image. Shrinking of image of size 320 x 280 pixels means the integro-differential operator (Eq. 3.1) will be applied 10 times totally (5 times for each boundary) at 5 stages (on 1 image at a stage). This is done after the parameters are converted. The process is exactly the same in each stage, except the start stages of detection. Pupil detection starts at stage 1*, and iris detection starts at stage 3*.

- Stage 1: The smallest image is convolved with Gaussian filter of size 1 x 5 with σ = 0.5*. The darkest pixel near the centre of the image is taken as the centre of the pupil. In this image (of size 20 x 18 pixels), the radius of the pupil is estimated to be from 1 to 5 pixels*. Daugman's operator is applied and the centre x, y coordinates and the radius of the best fitting circle for the pupillary boundary are found.

- Stage 2: The coordinates of the previously detected boundary or boundaries (in steps 4 and 5) are translated (by multiplying x, y and r by $S_f = 2$) and refinement

Stage	r from	r to	x from	x to	y from	y to
1	$0+1$	$0+5$	darkest x	darkest x	darkest y	darkest y
2	$r_{p1}*2-2$	$r_{p1}*2+2$	$x_{p1}*2-1$	$x_{p1}*2+1$	$y_{p1}*2-1$	$y_{p1}*2+1$
3	$r_{p2}*2-2$	$r_{p2}*2+2$	$x_{p2}*2-2$	$x_{p2}*2+2$	$y_{p2}*2-2$	$y_{p2}*2+2$
4	$r_{p3}*2-2$	$r_{p3}*2+2$	$x_{p3}*2-2$	$x_{p3}*2+2$	$y_{p3}*2-2$	$y_{p3}*2+2$
5	$r_{p4}*2-6$	$r_{p4}*2+3$	$x_{p4}*2-2$	$x_{p4}*2+3$	$y_{p4}*2-2$	$y_{p4}*2+3$

Table 6.1: Translation process for pupil detection

values (see section 6.6) are added and/or subtracted to form the upper and lower boundaries of the search space parameters. Daugman's operator is applied to detect the pupillary boundary.

- Stage 3: For pupillary boundary detection, the process is identical as in Stage 2. For limbic boundary detection, the parameters of the pupil boundary at stage 3 are taken as a reference point to form the search space. Refinement values are added and/or subtracted to calculate the set of potential centre points in the search space. The boundaries of the relative size of the pupil to the iris are set by Daugman et al. [4] - iris radius is 1.25 to 10 times larger than the radius of the pupil.

- Stage 4 and 5: Same as Stage 2.

* The values were empirically estimated to balance between time efficiency and accuracy.

In Table 6.1 and Table 6.2 the whole stage-to-stage parameter translation process, with the default refinement parameters for the search space of pupil and iris detection, is shown. To any cell in the table, further refinement can be done by adding or subtracting integer values.

Stage	r from	r to	x from	x to	y from	y to
3	$r_{p3} * 1.25 + 1$	$r_{p3} * 10 + 0$	$x_{p3} - 2$	$x_{p3} + 2$	$y_{p3} - 2$	$y_{p3} + 2$
4	$r_{i3} * 2 - 4$	$r_{i3} * 2 + 4$	$x_{i3} * 2 - 2$	$x_{i3} * 2 + 2$	$y_{i3} * 2 - 2$	$y_{i3} * 2 + 2$
5	$r_{i4} * 2 - 6$	$r_{i4} * 2 + 4$	$x_{i4} * 2 - 4$	$x_{i4} * 2 + 2$	$y_{i4} * 2 - 4$	$y_{i4} * 2 + 2$

Table 6.2: Conversion process for iris detection

6.6 Search space refinement

The search space for each stage and each boundary can be further enlarged or shrunk by changing the default search space refinement values (Table 6.3) (also see 6.7). Enlarging the search space will inevitably increase the detection time, and possibly, improve the detection rate as more circles are being analysed. In contrast, shrinking the search space might cause an incorrect detection of one or both boundaries.

Stage	PRF_l	PRF_u	PXF_l	PXF_u	PYF_l	PYF_u
1	1	5	0	0	0	0
2	-2	2	-1	1	-1	1
3	-2	2	-2	2	-2	2
4	-2	2	-2	2	-2	2
5	-6	3	-2	3	-2	3
Stage	IRF_l	IRF_u	IXF_l	IXF_u	IYF_l	IYF_u
3	1	0	-2	2	-2	2
4	-4	4	-2	2	-2	2
5	-6	4	-4	2	-4	2

Table 6.3: Search Space Refinement Values

6.7 Algorithm Diagram

Figure 6.3 is a diagram of the proposed algorithm. PRF_l and PRF_u (lower and upper search space refinement values for r_p), PXF_l and PXF_u (lower and upper search space refinement values for x_p), PYF_l and PYF_u (lower and upper search space refinement values for y_p) are the search space refinement values for pupillary boundary and IRF_l, IRF_u, IXF_l, IXF_u, IYF_l, IYF_u for the limbic boundary. The parameters of the detected pupillary boundary are r_p, x_p, y_p, for the limbic boundary - r_i, x_i, y_i. The formed search space ranges are Sr, Sx, Sy for radius, x-axis potential centre points and y-axis potential centre coordinates points, respectively.

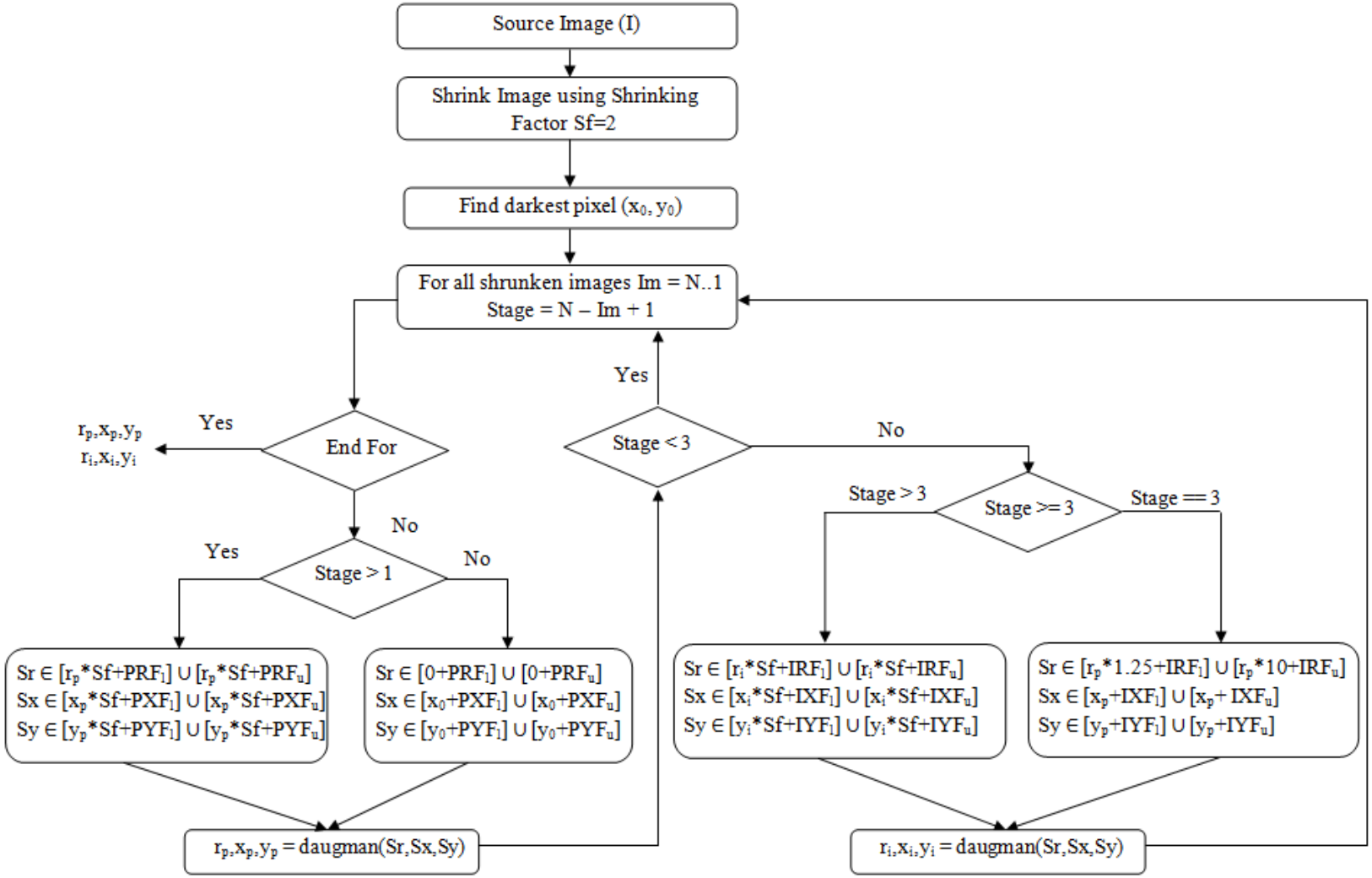


Figure 6.3: Algorithm Diagram

6.8 Normalization

A remapping formula is used by Masek et al. [10] and Shamsi et al. [15] to unwrap the iris. If the centre of the pupil is used as a reference point, to pick equally spaced points on the radial line, the following formula is used to give the length of the line connecting the centre and the limbic boundary at an angle θ

$$r\prime = \sqrt{\alpha}\beta \pm \sqrt{\alpha\beta^2 - \alpha - r_I^2} \tag{6.2}$$

With:

$$\alpha = o_x^2 + o_y^2 \tag{6.3}$$

$$\beta = \cos(\pi - \arctan(\frac{o_x}{o_y}) - \theta) \tag{6.4}$$

The displacement of the centre of the pupil relative to the centre of the iris is given by o_y, o_x. The distance between the edge of pupil and the edge of iris at an angle θ is $r\prime$ and the radius of the iris is r_I.

After a careful consideration of Eq. 6.2, it has been noticed, that in the special case, when there is no displacement between pupil and iris centres (the circles are concentric), for any value of β, when solved with:

$$\alpha = o_x^2 + o_y^2 = 0^2 + 0^2 = 0$$

$$\begin{aligned} r\prime &= \sqrt{\alpha}\beta \pm \sqrt{\alpha\beta^2 - \alpha - r_I^2} \\ &= \sqrt{0}\beta \pm \sqrt{0\beta^2 - 0 - r_I^2} \\ &= 0 \pm \sqrt{0 - r_I^2} \\ &= \pm\sqrt{-r_I^2} \end{aligned}$$

the solutions are imaginary numbers, which lie on the vertical axis of the complex plane. There might be other cases when the remapping formula will not adequately

convert the iris tissue. The formula has not been further analysed, but a new one has been derived from the representation of a circle in polar form [17]:

$$r = r_0 \cos(\theta - \phi) \pm \sqrt{a^2 - r_0^2 \sin^2(\theta - \phi)} \tag{6.5}$$

> where a is the radius of the circle, r_0 is the distance from the origin (the pupillary boundary) to the centre of the circle (the limbic boundary), and ϕ is the anticlockwise angle from the positive x-axis to the line connecting the origin to the centre of the circle. [17]

Let:

$$a = r_I \tag{6.6}$$

$$r_0 = \alpha = o_y^2 + o_x^2 \tag{6.7}$$

$$\beta = \cos(\theta - \phi) \tag{6.8}$$

$$\phi = \arctan(\frac{o_y}{o_x}) \tag{6.9}$$

The equation is simplified to:

$$\begin{aligned} r\prime &= \sqrt{\alpha}\beta \pm \sqrt{r_I^2 - \alpha \sin^2(\theta - \phi)} \\ &= \sqrt{\alpha}\beta \pm \sqrt{r_I^2 - \alpha(1 - \cos^2(\theta - \phi))} \\ &= \sqrt{\alpha}\beta \pm \sqrt{r_I^2 - \alpha + \alpha \cos^2(\theta - \phi)} \\ &= \sqrt{\alpha}\beta \pm \sqrt{r_I^2 - \alpha + \alpha\beta^2} \end{aligned} \tag{6.10}$$

To prevent non-iris data (artefacts) to be included in the normalized image, the points which lie on the pupillary and the limbic boundaries are discarded, following Daugman's rubber sheet model.

6.9 Graphical User Interface

A simple graphical user interface, has been developed to allow the users to execute the iris detection and normalization algorithms more easily. The GUI (Figure 6.4) allows an image to be loaded, the iris segmentation algorithm to be applied on the selected image, the search space refinement values to be changed (to improve the detection rate, when the detection fails), and the detected region to be normalized. The execution times for both algorithms are displayed.

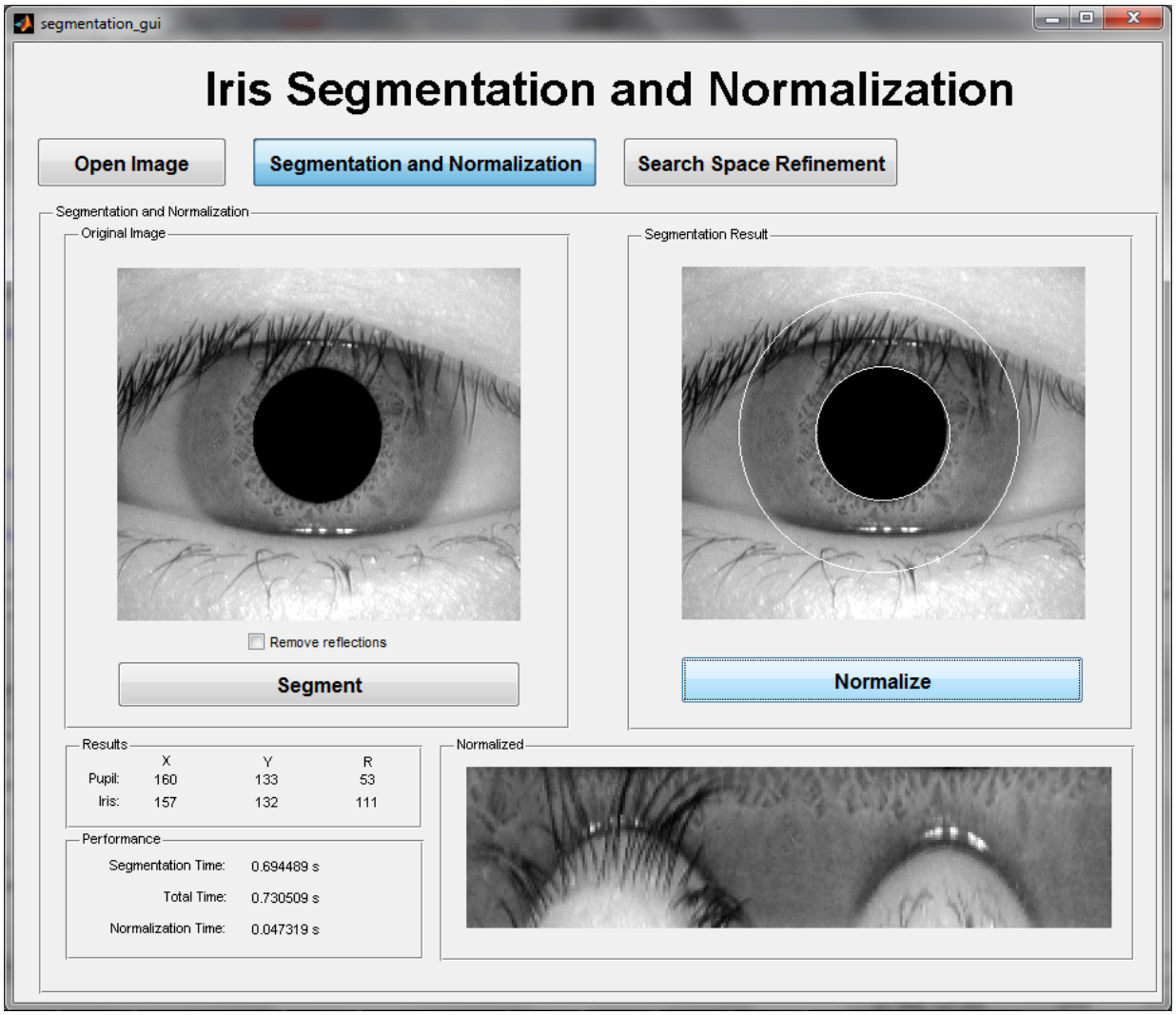


Figure 6.4: Algorithm Diagram

CHAPTER 7

IMPLEMENTATION

The iris detection and normalization algorithms are fully implemented in MATLAB - a procedural language, combining an effective programming structure with a bevy of predefined mathematical commands [14]

MATLAB is extremely effective, when performing computations on matrices - the main data structure. It comes with "toolboxes" - tools to implement more efficiently image processing, digital signal processing and various other algorithms [14].

7.1 Image Processing Toolbox

The main reason this was the chosen implementation language, is the tools in the image processing toolbox MATLAB provides - set of predefined commands for image editing and preprocessing. The ones used in the development of the algorithms are:

- imread() - reading and image from file
- im2gray() - conversion to Gray scale colour system
- imadjust() - contrast improvement
- imfill() - hole filling algorithm used to remove specular reflections
- imresize() - image resizing

- imfilter() - applying special filter on the image - for example Gaussian blur
- imshow() - displaying an image on the screen

All images are represented as matrices, where every cell of the matrix represents a pixel in the image. The manipulation is very easy, because values can be directly accessed by specifying the indices in the matrix. The origin of the coordinate system is the top left corner, where the x-axis stretches horizontally to the right and the y-axis stretches vertically to the bottom.

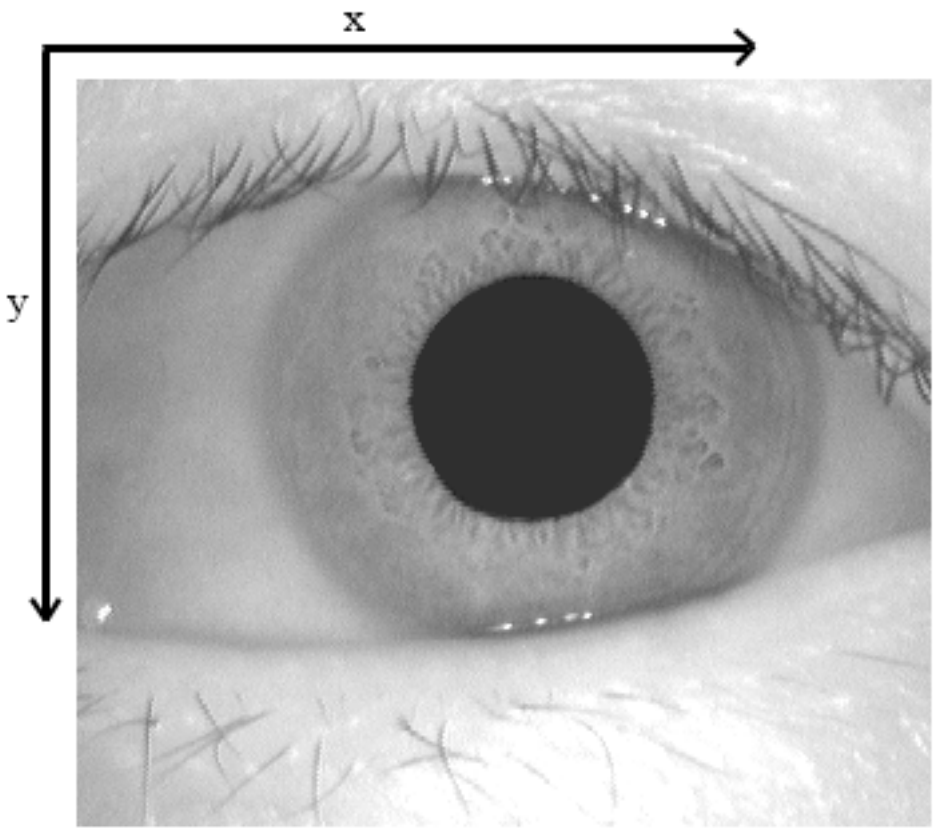


Figure 7.1: Coordinate System

7.2 Structure

The project has a modular structure and the iris detection (segmentation), normalization and evaluation modules, are put in separate folders. Because of this structure, the different modules can be easily incorporated in other projects. The graphical user interface is designed using MATLAB's GUI design environment (GUIDE) and joins the iris detection and segmentation modules.

CHAPTER 8

PROJECT MANAGEMENT

The project was built using incremental software development method. After the conducted research on the methods used for iris detection, prototypes of the product were built, tested and verified. The results of the prototypes led to the selection of a suitable problem solution (see chapter 6). As a consequence, more functionality was built on top of one of the chosen prototype.

8.1 Planned Gantt Chart

Figure 8.1 shows the planned Gantt Chart. The plan was not strictly followed due to unforeseen circumstances.

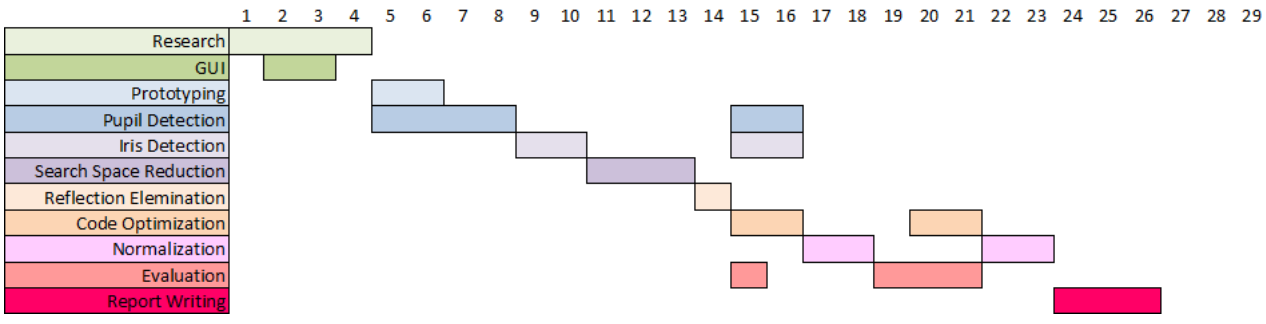


Figure 8.1: Planned Gantt Chart (Duration is in weeks)

8.2 Actual Gantt Chart

Figure 8.2 shows the actual Gantt Chart, with the delay in the project.

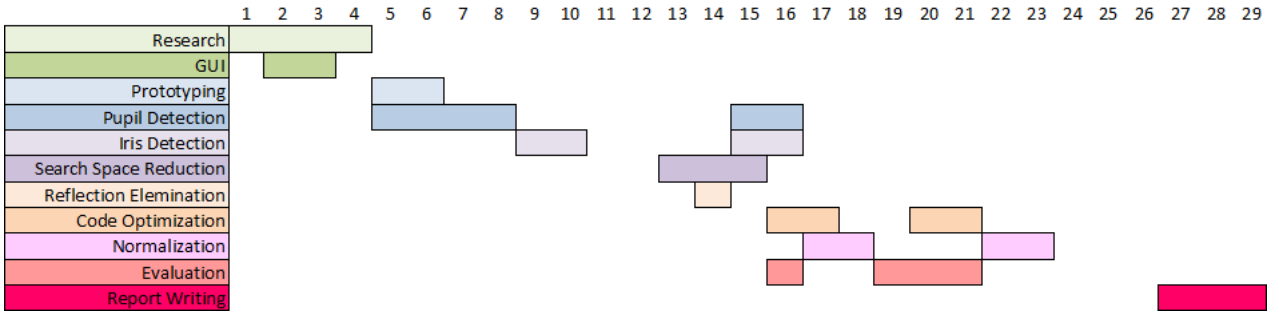


Figure 8.2: Actual Gantt Chart (Duration is in weeks)

CHAPTER 9

EVALUATION

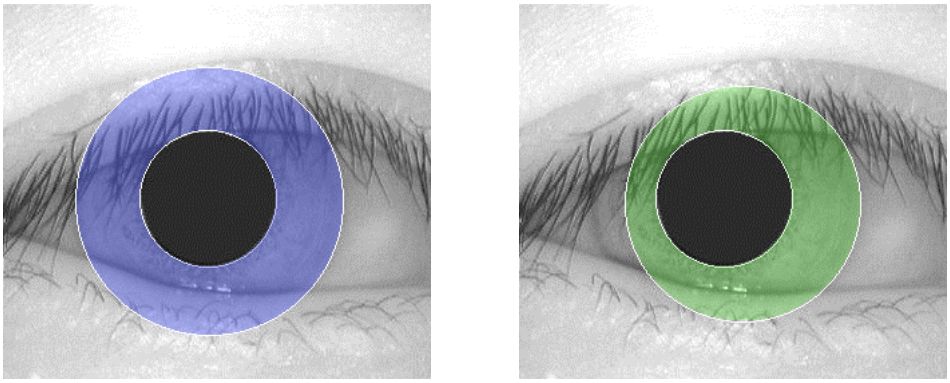
To evaluate pupil and iris detection rate, two methods for evaluation are used - a newly proposed quantitative method and a subjective method (correct/incorrect detection), used in most iris detection papers and journals. The algorithms are evaluated on a PC with 1.66 GHz Intel Core2Duo processor and 2GB of physical memory, running 64bit version of MATLAB R2009b.

9.1 Quantitative Evaluation

The detection rate is measured by Jaccard's similarity coefficient (Eq. 9.1), which measures the similarity between sets - the set of pixels in ground truth (manually estimated) iris boundaries and the computed boundaries using the proposed method in chapter 6.

$$J(G, C) = \frac{|G \cap C|}{|G \cup C|} \tag{9.1}$$

A test set of 30 randomly selected images from CASIA v1.0 was selected. The circles for pupillary and limbic boundaries were manually fit to isolate the iris tissue. The ground truth set (G) is the set difference between the pixels in iris and pupil, which were manually estimated (see Figure 9.1(a)). The computed set (C) is the set difference between the pixels in iris and pupil, computed by the developed iris detection algorithm (see Figure 9.1(b)).



(a) Ground Truth (Blue Region) (b) Computed (Green Region)

Figure 9.1: Ground Truth and Computed Sets

The results of the algorithm employed by Masek et al. [10] and the proposed approach are shown in Figure 9.2 and Table 9.1. A comparison chart between the results of the algorithms is given. Two identical sets will have similarity coefficient of 1.0. Further work will try to establish the relation between the similarity coefficient and the Hamming distance, used in the matching stage.

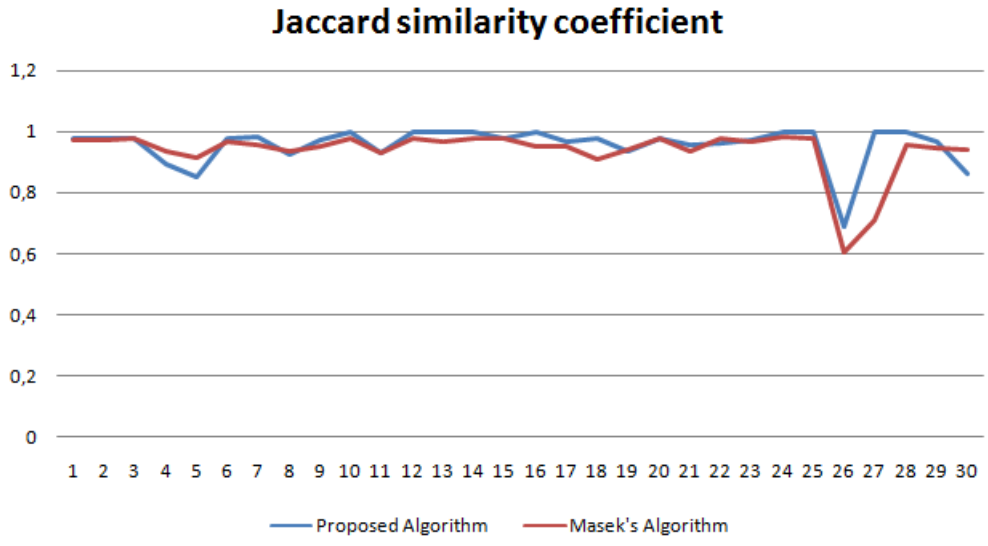


Figure 9.2: Jaccard similarity coefficient

Jaccard coefficient	Proposed Algorithm	Masek's Algorithm
Average	0.96	0.94
Min	0.69	0.60
Max	1.00	0.98
Standard Deviation	0.06	0.08

Table 9.1: Jaccard similarity coefficient comparison

9.2 Subjective Evaluation

A subjective evaluation of the proposed iris segmentation method was performed on a set of 150 images from CASIA V1.0 (without reflections) and a set of 150 images from CASIA V3.0 (with specular reflections) databases. The results for pupillary and/or limbic boundaries detection, with and without reflection removal, are shown in Table 9.2.

Reflection Elimination	CASIA V1.0 Pupil and Iris	CASIA V3.0 Iris	CASIA V3.0 Pupil
No	88%	85%	92%
Yes	-	93%	99%

Table 9.2: Subjective Evaluation Results

9.3 Time efficiency

The times of detection of Masek's iris detection algorithm [10] and the proposed method are shown in Figure 9.3. The achieved execution times are listed in Table 9.3. The results of the proposed algorithm entirely satisfy the project's initial requirements.

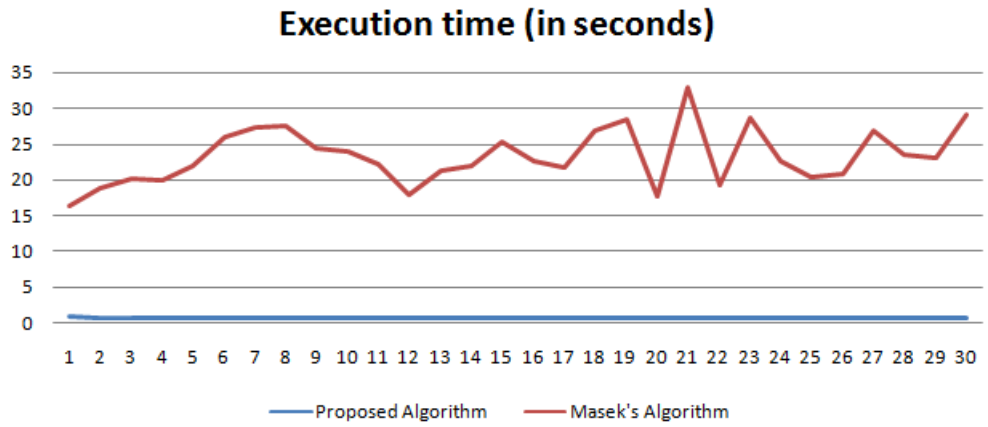


Figure 9.3: Execution Time

Time (seconds)	Proposed Algorithm	Masek's Algorithm
Average	0.66	23.42
Min	0.62	16.46
Max	0.86	33.00
Standard Deviation	0.04	3.92

Table 9.3: Execution Times

9.4 Number of circles and pixels analysed

To illustrate the significance of a search space reduction algorithm, a comparison of the number of analysed circles and the number of pixels accessed between Daugman's algorithm (with the boundaries set by Masek et al. [10] - see section 5.2) and the proposed approach is shown in Figure 9.4, Figure 9.5 and Table 9.4. The number of processed "elements" in the developed algorithm is significantly lower, than if the boundaries of the search space were fixed and estimated manually.

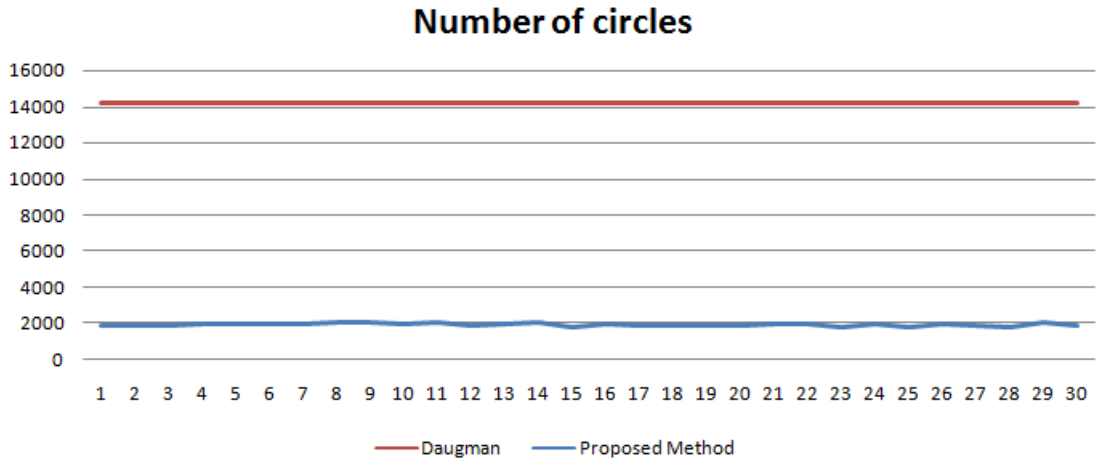


Figure 9.4: Number of analyzed circles

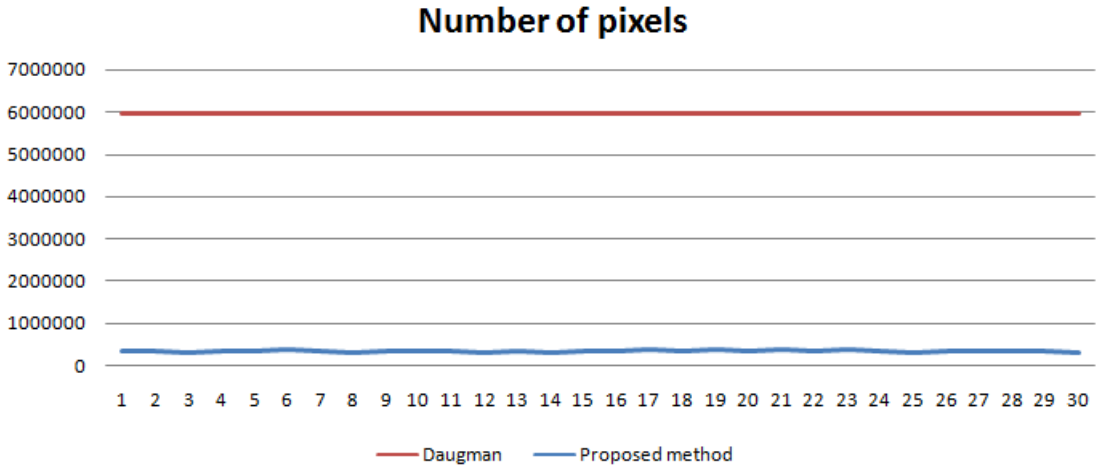


Figure 9.5: Number of pixels taken for computations

	Circles		Pixels	
	Proposed	Daugman	Proposed	Daugman
Average	1944	14157	354774	5953200
Min	1824	14157	315548	5953200
Max	2054	14157	394734	5953200
Standard Deviation	64	0	20628	0

Table 9.4: Circles/Pixels analysis

9.5 Disadvantages of the proposed method

There are two main disadvantages of the proposed algorithm. They could be resolved in future work, to improve the detection accuracy.

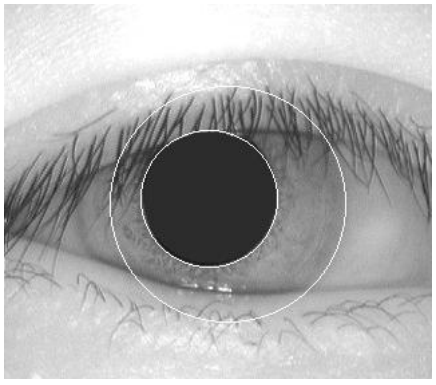


Figure 9.6: Inaccurate iris detection, caused by eyelid occlusion

- A major drawback of the algorithm is that it lacks the ability to detect eyelids and eyelashes in the images. A presence of eyelid occlusion and/or eyelashes in the iris region can cause two major problems - incorrect iris detection (Figure 9.6) or a significant area in the normalized image, which is occupied by artefacts (Figure 5.2(b)). Incorrect boundary detection and/or artefacts in the normalized image significantly increase the probability of false rejection, if the algorithm is used in an iris recognition system.

- Further complications, which can have impact in iris recognition systems, can be caused by the shape of iris and pupil - elliptical instead of circular. To improve the

accuracy, the algorithm have to be modified to search over an elliptical path instead of circular. Figure 9.7 illustrates a detection of circular pupillary boundary, when the shape of the pupil is an ellipse - the detected pupillary boundary is larger than the pupil and when the region is normalized, usable data will be omitted. There is another case, when the detected pupil region is smaller than the pupil, causing artefacts to be present in the normalized image.

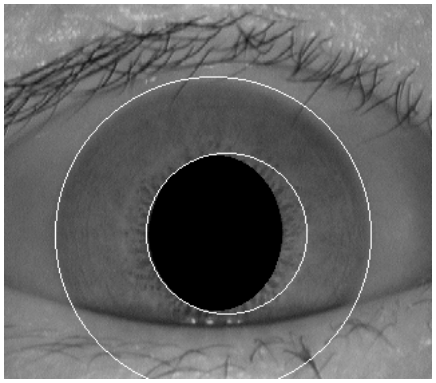


Figure 9.7: Inaccurate pupil detection, caused by the elliptical shape of the pupil

CHAPTER 10

SUMMARY

The goal of the project was to develop an accurate and time efficient iris detection and normalization algorithms, which can both be used in iris recognition systems. Testing, validation and evaluation of the developed algorithm were performed on images of CASIA iris image databases V1.0 and V3.0, with and without specular reflections in the images, respectively.

An iris detection algorithm must be able to automatically identify pupillary (between pupil and iris) and limbic (between iris and sclera) boundaries. The boundaries usually are circles, therefore a circle detection algorithms were researched. It was necessary, they do not only meet project's requirements, but offer favourable circumstances to include a piece of originality in the development of the algorithm.

During the research, Daugman's Integro-Differential Operator, has been identified as the best solution to the problem, because it allowed improvement and code optimization. The way the proposed method works, is by applying the operator to limited number of circles in dynamically estimated boundaries instead of applying it to unnecessary large number of circles in boundaries, which are manually estimated for the database used. Daugman's operator searches for the maximal difference between the mean values of the pixels lying on the circumferences of the circles in images. To speed-up the algorithm even more, a midpoint circle algorithm is applied to generate the coordinates of the pixels, which lie on the circle circumference, using integer algebra only.

The developed solution is an improvement of John Daugman's algorithm in terms of accuracy (in images with reflections) and time efficiency. It fully meets the requirements, which were set at the beginning of the project.

The normalization algorithm is based on Daugman's Rubber-Sheet model, which converts the isolated area to a dimensionless coordinate system by Cartesian to non-concentric polar representation. The Rubber-Sheet model accounts for pupil dilation or constriction and size inconsistencies. A remapping formula was developed for this purpose.

Future versions of the algorithm could detect elliptical iris and pupil boundaries (instead of circular), implement eyelid detection and eyelash extraction algorithms to reduce artefacts in the normalized templates. This will reduce the false rejection rate in iris recognition systems.

APPENDIX A

CD CONTENTS

- algorithm/ - the MATLAB files of the algorithm
- CASIA/ - CASIA Iris Image Database
- CASIA_TEST/ - the test images subsets
- IrisDetection_Report.pdf - electronic copy of this report
- evaluation.mat = the evaluation data (MATLAB Data File)
- Evaluation.xls - the evaluation spreadsheet

APPENDIX B

HOW TO RUN

The algorithm can be run in two ways - through MATLAB's command line or through the GUI.

B.1 Command Line

B.1.1 Iris Detection

1. To run the segmentation algorithm through the command line, navigate to folder "segmentation".

2. Load an iris image by running "*img=imread('path_to_image');*"

3. Detect pupillary and limbic boundaries by running "*[pupil, iris, time] = segment(img, 0, [], [])*". This command will return the parameters of pupillary and limbic boundaries. The format of the command is *segment(image, remove_reflections, pupil_search_space, iris_search_space).* pupil(1),pupil(2) and pupil(3) are x, y and r parameters of the pupillary boundary and iris(1),iris(2) and iris(3) are x, y and r parameters of the limbic boundary.

4. You can display the result by running "*figure, imshow(draw_circle(draw_circle(img, pupil(1), pupil(2), pupil(3), 255), iris(1), iris(2), iris(3), 255));*"

B.1.2 Normalization

1. To run the normalization algorithm through the command line, navigate to folder "normalization".

2. Once the boundaries are detected, you can normalize the detected region by running "*normalized = normalize(img, pupil(1), pupil(2), pupil(3), iris(1), iris(2), iris(3), 64, 256);*". This will produce a normalized image with radial resolution 64 and angular resolution 256.

3. To display the result run "*figure,imshow(normalized);*"'

B.1.3 Evaluation

1. To run the evaluation test navigate to folder "evaluation".

2. Import the evaluation data into the runspace by running "*importdata('path_to_datafile')*"

3. To get the Jaccard coefficient between the ground truth and the results, run "*coeff = evaluate(result(1:30,1:3),result(1:30,4:6),gt(1:30,1:3),gt(1:30,4:6))*"

4. To get the Jaccard coefficient between the ground truth and Masek's algorithm results, run "*coeff = evaluate(masek(1:30,1:3),masek(1:30,4:6),gt(1:30,1:3),gt(1:30,4:6))*"

5. The average value can be calculated by running "*mean(coeff)*";

B.2 GUI

To run the graphical used interface, navigate to the main folder and run "*segmentation_gui()*".

LIST OF REFERENCES

[1] Find Biometrics. Biometrics characteristics. Available on the web at: http://www.findbiometrics.com/.

[2] St. Lukas Cataract and Laser Institute. Eye anatomy. Available on the web at: http://stlukeseye.com/anatomy/iris.html.

[3] John Daugman. Biometric personal identification system based on iris analysis. U.S. Patent No. 5,291,560 issued March 1, 1994.

[4] John Daugman. How iris recognition works. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 1, pages I33–I36, 2002.

[5] John Daugman. Probing the uniqueness and randomness of iriscodes: Results from 200 billion iris pair comparisons. In *Proceedings of the IEEE*, volume 94, issue 11, pages 1927–1935, November 2006.

[6] Center for Biometrics and Security Research. Casia iris image databases. Available on the web at: http://biometrics.idealtest.org/.

[7] A. P. Godse. *Computer Graphics*. Technical Publications Pune, #1 Amit Residency, 412, Shaniwar Peth, Pune - 411 030, India, 2007.

[8] Nguyen Van Huan and Hakil Kim. A novel circle detection method for iris segmentation. 2008.

[9] Motorola Limited. An introduction to biometrics, August 2006. Available on the web at: http://www.motorola.com/web/Business/Solutions/Business\%20Solutions/Biometrics/Biometrics\%20Identification/_Documents/Static\%20Files/INTRODUCTION_TO_BIOMETRICS_New.pdf.

[10] Libor Masek. Recognition of human iris patterns for biometric identification. 2003.

[11] M.A. Mohammed. Automated algorithm for iris detection and code generation. *Computer Engineering & Systems, 2009. ICCES 2009. International Conference on*, pages 475–781, 2010.

[12] H. B.N. Soltanian-Zadeh M.S. Araabi Hosseini. Pigment melanin: Pattern for iris recognition. In *Instrumentation and Measurement, IEEE Transactions on*, volume 59, issue: 4, pages 792–804, April 2010.

[13] Tanmay Rajpathak. Eye detection using morphological and color image processing. *Florida Conference on Recent Advances in Robotics*, 2009.

[14] Darren Redfern and Colin Campbell. *MATLAB 5 Handbook*. Springer-Verlag, Boston, 1998.

[15] Mahboubeh Shamsi. A new accurate technique for iris boundary detection. *WSEAS Transactions on Computers*, volume 9, issue: 6, June 2010.

[16] Biometric Solutions. Iris recognition. Available on the web at: `http://www.biometric-solutions.com/solutions/index.php?story=iris_recognition`.

[17] Wikipedia. Circle, polar coordinates. Available on the web at: `http://en.wikipedia.org/wiki/Circle#Polar_coordinates`.