7

Solutions for Iris Segmentation

Milena Bueno Pereira Carneiro, Antônio Cláudio P. Veiga,
Edna Lúcia Flôres and Gilberto A. Carrijo
*Federal University of Uberlândia – Department of Electrical Engineering,
Brazil*

1. Introduction

The growing concern with security and access control to places and sensitive information has contributed for the increased utilization of biometric systems. Biometry is the name given to the techniques used to recognize people automatically through physical and behavioural characteristics of the human body such as those found on the face, fingerprint, hand geometry, iris, signature or voice. From all the biometric options, iris recognition deserves special attention as the iris contains a huge and unique richness of characteristics, which do not change over time and enables the construction of extremely reliable and accurate systems.

The iris recognition process is relatively complex and involves several stages of processing as illustrated in Figure 1. The first stage corresponds to the localization of the region of the iris on the image of the eye, which also involves the extraction of the regions corrupted by the superior and inferior eyelids and eyelashes.

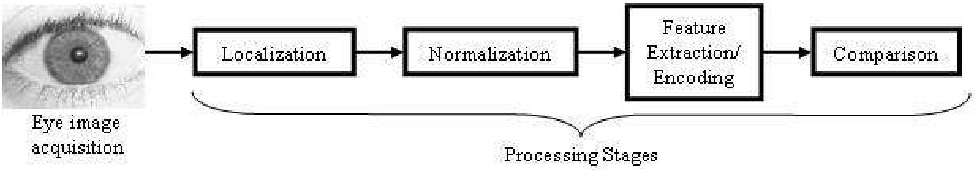


Fig. 1. Processing stages of an iris recognition system.

Once the region of the iris has been located, it must go through the normalization process. This stage is responsible for solving the problem of dimensional inconsistencies that are generally provoked by the variation of the distance between the individual and the image capture device and also by the variation of the size of the pupil due to varying levels of luminosity of the environment. The most utilized normalization method is called the "rubber sheet model" which was first suggested by Daugman (Daugman, 1993). In this model, all the annular region of the iris is uniformly sampled, in the radial direction as well as the angular direction, and is represented through the polar coordinate system, consequently generating a rectangular image.

After the normalization, the information from the iris is extracted and encoded so that the comparison of the images can finally be possible.

This chapter will focus on the first processing stage that is the segmentation (or localization) of the iris region in an eye image. The efficiency of the localization stage is essential to the success of the iris recognition system given the fact that error rates will increase if regions that do not belong to the iris are coded and processed.

In the next section, two traditional iris segmentation algorithms and a method to detect eyelashes and eyelids are described, implemented and some experimental results are collected.

In section 3, we will propose a new method for iris segmentation. The proposal is based on the application of Memetic Algorithms to detect the circles which define the iris region. The implementation details of the new method are described and the experimental results are presented.

Section 4 presents an analysis of the effects of using severe compressed images. The performance of all the segmentation algorithms described previously will be measured for images with different levels of compression. Two different compression techniques will be employed.

Images from the public iris image database UBIRIS (Proença & Alexandre, 2005) were used to execute the experimental tests.

2. Traditional methods for iris segmentation

The pupil border and the external border of the iris, that together define the iris region, present a near elliptical contour. Nevertheless, the most traditional researches on iris recognition approximate them to circles with minimal loss in performance. Consequently, circle detection algorithms are usually applied to perform the iris localization.

Investigating the iris localization methodologies in the literature, it is possible to identify two major strategies on which those methodologies are based. Some methods are template-based and usually involve the maximization of some equations. Otherwise, they are boundary-based and demand the construction of an edge map for a later application of some geometric form-fitting algorithm.

Two methods were evaluated, each of them representing one of the possible strategies cited.

The most traditional and widely used template-based method is the Integro-Differential Operator proposed by John Daugman (Daugman, 1993).

The most important boundary-based methods apply the Circular Hough Transform to find circles in an edge map, which was first suggested by Wildes (Wildes, 1997). This localization methodology was used, with several small variations, in many works such as (Huang et al., 2004; Ma et al., 2004; Masek, 2003) and many others.

The segmentation of the iris region also includes the detection and the exclusion of the eyelids and eyelashes interferences. The method proposed by Libor Masek (Masek, 2003) to exclude those interferences was also evaluated.

The main features of the two iris segmentation methods and the eyelids and eyelashes detection method applied are briefly explained in the following subsections.

2.1 Circular Hough transform

The Circular Hough Transform (CHT) (Gonzalez & Woods, 2002) was proposed by Paul Hough in 1962. This technique is able to recognize the circles present in an image and can be used to obtain the parameters that define the circle that represents the pupil border and the circle that represents the external iris border. These parameters are the radius and the coordinates of the circle center.

The first step is to convert the gray-scaled eye image into a binary edge map. The construction of the edge map is accomplished by the Canny edge detection method (Canny, 1986) with the incorporation of gradient information. The details of the Canny operator will not be included to this paper. It is important to know that the application of the Canny operator demands the adjustment of some parameters. After the adjustment, the same parameter values were used in all experimentations.

The Hough procedure requires the generation of a vote accumulation matrix with the number of dimensions equal to the number of parameters necessary to define the geometric form. For a circle, the accumulator will have 3 dimensions.

In the CHT, each edge pixel with coordinates (x,y) in the image space is mapped for the parameters space determining two of the parameters (for example, x_c and y_c) and finding the third one (for example, r) which resolves the circumference equation given in (1).

$$(x - x_c)^2+(y - y_c)^2 = r^2 \tag{1}$$

As a result, the point with coordinates (x_c, y_c, r) is obtained in the parameters space which represents a possible circle in the image.

At each set of parameters obtained (x_c, y_c, r), the value of the accumulator in this position A(x_c, y_c, r) is incremented, i.e., a vote is attributed to that position. When all the pixels have been processed, the highest values of the accumulator A (i.e. the position that has received the highest number of votes) will indicate the parameters of probable circles in the image.

2.2 Integro-differential operator

John Daugman (Daugman, 1993) proposed an integro-differential operator to locate the circular regions of the iris and the pupil. This operator is defined by the equation (2).

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

Where I(x,y) is the image of the eye, r is the search radius, $G_\sigma(r)$ represents a Gaussian smoothing function and s is the contour of the r radius circle with the center in (x_0, y_0).

The operator searches for the circular path where there is the greatest change in pixel values when there is a variation of the radius and the x and y coordinates from the center of the circle. The operator is applied iteratively with the amount of smoothing being progressively reduced in order to attain precise localization.

Although the iris search results depend on the pupil search, it is not possible to assume that the iris external circle has the same center of the pupil. Therefore, the three parameters that define the circle of a pupil must be estimated separately from the ones that define the iris.

2.3 Detection of eyelids and eyelashes

A procedure suggested by Libor Masek (Masek, 2003) was used to perform the detection of the region of the iris covered by the eyelids and eyelashes. To isolate the eyelids, it was assumed that their edges could be approximated by a line segment. The first step is to find a line that corresponds to the edge of the superior eyelid and one that corresponds to the inferior eyelid. To achieve this, the Linear Hough Transform was used. A second line is then drawn horizontally intercepting the first at the edge point of the iris closest to the pupil. This procedure is done for both the superior and the inferior eyelids. The region above the superior eyelid line and the region below the inferior eyelid line are excluded.

It may happen that, in some images, there is no occlusion of the iris by the eyelids. Therefore, if the maximum value in the Hough space is smaller than a predetermined threshold, no line is identified, which represents a non occlusion. Moreover, a line is only considered when it is found out of the pupil region and in the iris region.

To isolate the eyelashes a threshold determination technique is used, considering that in the group of images used the eyelashes are in general a little darker when compared to the rest of the image. Consequently, all of the pixels in the image darker than the threshold established are considers to be pixels which belong to the eyelashes and are consequently excluded.

2.4 Experimental results

There are presently several public and freely available iris image databases for biometric purposes. The majority of them incorporates few types of noise, almost exclusively related with eyelid and eyelash obstruction. The UBIRIS database (Proença & Alexandre, 2005) tries to simulate noncooperative imaging conditions. Its capture conditions produced images with very heterogeneous characteristics regarding focus, motion blur, contrast, brightness, as well as iris occlusions by eyelids or eyelashes and specular and lighting reflections. Mainly because of that, the UBIRIS database was chosen to be used in this work for the experimental tests, aiming to approach to a more realistic situation which makes the iris localization challenging. It was used a total of 1201 gray scale eye images with resolution 200x150.

The two iris segmentation methods and the eyelashes and eyelids detection method were applied to the images from the UBIRIS database. When the circular Hough transform was applied to the 1201 images, it obtained success in localizing the iris region in 1119 images, which means an efficiency of 93.17% and an error rate of 6.83%.

The integro-differential operator was able to correctly localize the iris region in 1135 images, which means an efficiency of 94.50% and an error rate of 5.5%.

The algorithm used to detect the eyelashes and eyelids interferences presented an efficiency of 95.25% (1144 images) and an error rate of 4.8%.

The classification of the results as correct or not correct was manually done by a visual inspection of the result of the application of each method in each image.

3. Proposed method for iris segmentation

We propose a method based on memetic algorithm to localize the iris region. In this section, the fundamentals of the proposal are described and the utilization of evolutionary algorithms is justified.

As well as the CHT, the proposed method demands that the original image passes through an edge detection algorithm. There are several edge detection techniques in the literature and, as well as in the implementation of the CHT, the Canny method (Canny, 1986) was employed.

It is necessary to obtain a binary edge image as that shown in Figure 2. In this figure, the white pixels represent edge pixels.

As suggested by Wildes (Wildes, 1997), only vertical edges were detected for external iris border localization, whereas vertical and horizontal gradients were equally weighted for pupil localization.

From Figure 2, one notices that the pupil and the iris contours are apparent. Suppose that, using a geometric figure of a circle (imaginary circle), one wishes to overlap, on the edge

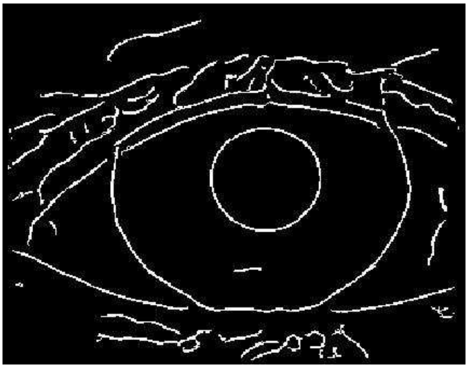


Fig. 2. Eye image after the application of the Canny edge detection algorithm.

image, the highest quantity of white pixels (edge pixels) possible. It is intuitive that, to reach this objective, the imaginary circle should be plotted onto the pupil or on the external border of the iris.

From this principle, it is possible to find the two desired circles through a scanning on the edge image. This scanning consists of varying the parameters of the imaginary circle into the image limits and, for each set of parameters (x_c, y_c, radius), to plot the corresponding circle on the edge image and to count the quantity of edge pixels that are overlapped. Thus, in the end of the scanning, the imaginary circle that has overlapped more edge pixels, is considered to be the contour of the pupil or the iris.

The search for each one of the desired circles must be carried out separately. The images of the database used (UBIRIS (Proença & Alexandre, 2005)) have a resolution of 200 pixels along the horizontal direction (coordinate x) and 150 pixels along the vertical direction (coordinate y). For those images, it can be considered that the radius of the pupil ranges from 5 to 36 pixels while the radius of the iris ranges from 40 to 71 pixels. Hence, to implement a complete scanning, the coordinate x_c must vary from 1 to 200, and for each assumed x_c, the coordinate y_c must vary from 1 to 150. Moreover, for each pair of center coordinates (x_c, y_c) the radius must vary from 5 to 36 for a pupil search or from 40 to 71 for an iris search.

Analyzing this procedure, one observes that, to find the external border of the iris, it is necessary to consider 200 * 150 * (5 - 36) = 930.000 imaginary circles, and to find the pupil, it is necessary to consider 200 * 150 * (40 - 71) = 930.000 imaginary circles. Under these conditions, this procedure is impracticable, as it demands intense computational processing.

This is a kind of problem for which there are many possible solutions and a vast search space. Some heuristic techniques are capable of approaching these problems. Here, Memetic Algorithms were employed to find the parameters of the iris and pupil circles in an adequate time, making the process less computationally demanding.

3.1 Memetic algorithms description

The called Memetic Algorithms (MAs) combine concepts of “local search” with the other operators typically used on Genetic Algorithms (GAs) that are inspired on Darwin's theory of evolution. Darwin established that, in a biological evolutionary process, after many generations, the population evolves according to the principles of natural selection and the survival of the fittest. Memetic algorithms take into consideration not only the ”genetic” evolution of individuals, but also a form of “cultural” evolution that is generally accomplished by a local search algorithms (Moscato, 2001).

The pseudo-code shown in Figure 3 contains the main steps of the computational implementation of the memetic algorithms. Firstly, in line 3, a population of individuals is generated randomly and stored in the variable *Pop*. The number of individuals in the population is *PopSize*. Each individual is represented by a chromosome, which is constituted by a code that represents a solution to the problem. All the generated individuals should be evaluated and a value is associated to each one of them in order to represent their fitness, which means, how well the solution fits the problem in question. The evaluation is carried out on line 5.

Lines 6 to 21 show the steps for creating the population of the next generation. The first step is the “Elitism””, which is used to improve the convergence of the algorithm. It consists of holding back the *n* best individuals in each generation.

To complete the new population, some individuals are selected for reproduction. The selection mechanism takes into consideration their fitness, so that an individual has a selection probability proportional to its fitness. Once two individuals are selected (lines 9 and 10) their genes are recombined through the crossover operators with a probability of “CrossoverProbability”. To achieve that, a “RandomNumber” between 0 and 1 is generated and if it is smaller than “CrossoverProbability” the crossover is executed and two new individuals are created (lines 13 and 14). The last genetic operator is the mutation, which alters the value of one or more genes of the chromosome chosen randomly with a small probability of “MutationProbability” (lines 15 to 17). Mutation increases the diversity of the population and assures that any point of the search space can be reached.

```
1 // MA Procedure
2
3 Pop = GenerateInitialPopulation(PopSize)
4 Repeat
5       CalculateFitness(Pop);
6       Elitism(n, Pop, NewPop);
7       i = n+1;
8       Repeat
9               Ind1 = Selection(Pop);
10              Ind2 = Selection(Pop);
11              NewPop(i) = Ind1;
12              NewPop(i+1) = Ind2;
13              If RandonNumber < CrossoverProbability
14                      [NewPop(i), NewPop(i+1)] = Crossover(Ind1, Ind2);
15              If RandonNumber < MutationProbability
16                      NewPop(i) = Mutation(NewPop(i));
17                      NewPop(i+1) = Mutation(NewPop(i+1));
18              i = i+2;
19      Until NewPop is complete
20      LocalSearch(NewPop);
21      Pop = NewPop;
22 Until stop criterion is satisfied
```

Fig. 3. Pseudo-code of the memetic algorithm procedure.

The two new individuals are added to the new population. When the new population is complete, a local search procedure is executed (line 20) for each individual. The local search algorithm evaluates the individuals in a neighbourhood defined by a mechanism of neighbourhood generation and if a fitter individual is found, it substitutes the previous individual and is added to the new population. A pseudo-code of a genetic algorithm can be obtained from the pseudo-code of Figure 3 by simply excluding line 20.

The whole procedure is repeated until the stop criterion is satisfied. The stop criterion can be the number of generations, the execution time or an acceptable value of fitness.

The application of MAs to the iris localization is described in the next section.

3.2 Implementation of the proposed method using memetic algorithms

An implementation of a memetic algorithm has been utilized to solve the aforementioned problem.

First, it is necessary to define a structure for the chromosomes, which will represent possible solutions to the problem. A 17-bit binary representation is used, and each gene of the chromosome is assigned a value 0 or 1; the first 6 bits (or genes) refer to the coordinate x_c for the circle center, the next 6 bits refer to the coordinate y_c of the center and the last 5 bits represent the radius of the circle.

The 6-bit structure makes possible to represent 64 different values of x_c and y_c. As mentioned before, the size of the images is 200 x 150, but, in general, the iris is positioned in the central part of the eye image, thus, it is not necessary to look for the centers of the iris and the pupil in the extremities of the image. Therefore, the coordinate x_c is considered to be in the range of 69 to 132 and the coordinate y_c is considered as being in the range of 44 to 107. The radius is represented by a 5-bit structure and so, 32 different values can be obtained. For the pupil, the 32 different values are between 5 and 36, and for the iris, they are between 40 and 71.

Another extremely important step is the definition of a fitness function that assigns a fitness score for each individual. The number of edge pixels that match the circle represented by the chromosome defines the fitness score for that individual. Thus, the greater the number of matched pixels, the fitter an individual is.

The algorithm must generate an initial population representing the initial search space. The generation of the initial population is carried out by randomly assigning one binary value to each gene of a chromosome.

Two populations of individuals have been generated, one consisting of candidates to represent the external border of the iris, and the other to represent the pupil border. These populations are processed independently.

Two common genetic operators have been used: one-point crossover and flip mutation. In the first case, a pair of individuals is selected from the population and their chromosomes are trimmed at a random position; this yields two tails that are swapped between the two chromosomes so that two new different individuals are generated. Mutation can be applied to every descendant after the crossover. It alters some genes of some chromosomes at random and with a low probability.

There are different crossover and mutation operators for binary chromosomes. The one-point crossover and flip mutation are the simplest operators found in literature. As they reached good results they were chosen in other to minimize the computation effort and the execution time.

Before proceeding to the next generation, a local search is performed to add the "cultural evolution". In the next section, the local search mechanism used to implement the memetic algorithm is described.

3.3 Implemented local search mechanism

Observation of the performance of the evolutionary part of the algorithm (with no local search) demonstrates that many times, solutions remain stuck at local minima for several generations. Graphically, it looks like the picture in Figure 4.

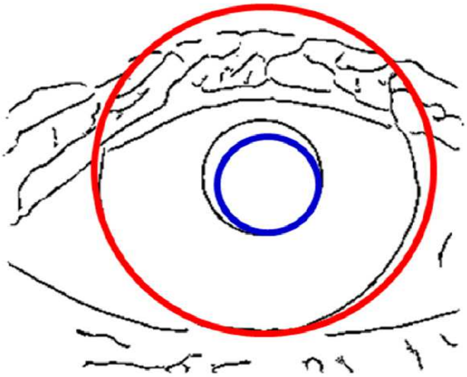


Fig. 4. The local minima issue that often occurs during the performance of a GA.

One notices that the imaginary circles, which represent the solution in question, are very close to an optimal solution, but the evolutionary algorithm could only generate an individual with these optimal characteristics after many generations. This is exactly where a local search becomes advantageous and accelerates the convergence.

The neighbourhood of the solution could be investigated by shifting the center of the circle to the right and to the left (x coordinate), upwards and downwards (y coordinate) and, for eachof these variations, the radius should be ranged. However, this form of local search is computationally expensive, especially when the population contains a great number of individuals.

For that reason, in order to lessen the computational effort of the local search, a strategy for reducing the neighbourhood has been proposed and implemented, which guides the coordinates of the circle. Visually, one notices that the solution in Figure 4 is such that the circle touches the pupil border in a certain extension, i.e. some pixels of the imaginary circle are coincident with pixels of the pupil border. This information might be useful to guide the coordinates of the circle.

The local search mechanism is structured in the pseudo-code shown in Figure 5. It is applied to all individuals of the population (line 2), and each individual contains the parameters of a circle. The circle can be divided into quadrants and two diagonal lines can be defined as shown in Figure 6.

```
// Procedure LocalSearch(NewPop)
For each individual of the population
    Repeat
        Draw circle represented by the individual on edge image;
        Discover each quadrant of the circle coincides with the highest...
        ... amount of edge pixels;
        If the highest amount of coincident edge pixels is in quadrant 1
            SelectedDiagonal = diagonal A;
        If the highest amount of coincident edge pixels is in quadrant 2
            SelectedDiagonal = diagonal B;
        If the highest amount of coincident edge pixels is in quadrant 3
            SelectedDiagonal = diagonal A;
        If the highest amount of coincident edge pixels is in quadrant 4
            SelectedDiagonal = diagonal B;
        i = 1;
        While i < = centerRange Do
            Shift the center along SelectedDiagonal;
            j = 1;
            While j < = radiusRange Do
                Vary radius;
                NewFitness = Calculate fitness of the new ...
                ...individual (new center and new radius)
                If NewFitness > Fitness of the original individual
                    Original individual = New individual;
                j = j+1;
            endWhile
            i = i+1;
        endWhile
    Until no improvement is detected during a iteration.
endFor
```

Fig. 5. Pseudo-code of the local search procedure.

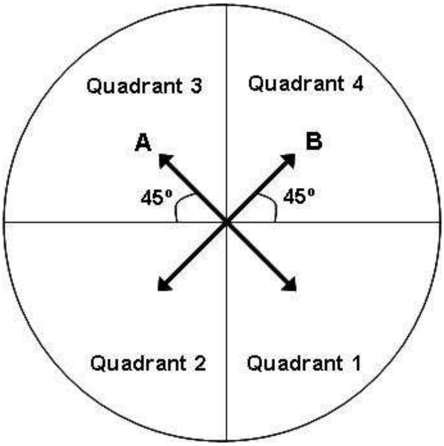
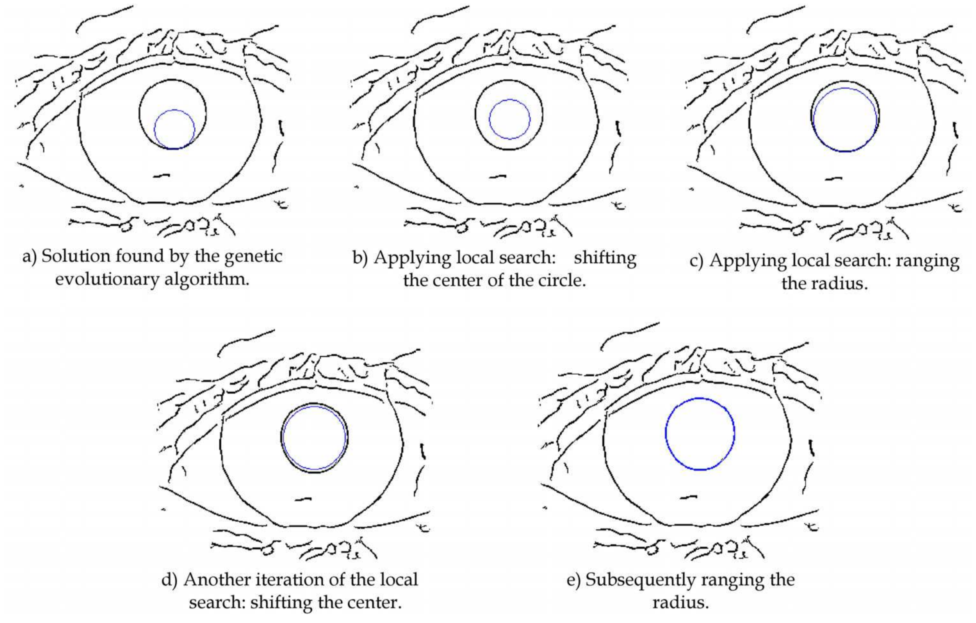


Fig. 6. Direction of the possible diagonal lines to be selected during the local search procedure.

To evaluate the fitness of an individual, the amount of edge pixels that coincide with the circle when it is plotted on the edge image is summed up. The local search algorithm verifies in which of the quadrants there have been the highest number of coincident pixels (line 5). If quadrants 1 or 3 have the highest amount of coincident pixels, the local search mechanism will vary the center coordinates x_c and y_c maintaining the center of the imaginary circle always along the diagonal line A (lines 7-8 and 11-12). If, on the other hand, the majority of the coincident pixels is in quadrants 2 or 4 the center of the imaginary circle must be shifted along the diagonal line B (lines 9-10 and 13-14). Therefore, to evaluate the neighbourhood, the center of the imaginary circle must be shifted a number of times equal to "centerRange" along the selected diagonal (lines 16 and 17), and, for each assumed position, the circle radius must also be ranged a number of times equal to "radiusRange" (line 19 and 20). For each situation, the new fitness must be calculated (line 21). If the new fitness is higher than the original fitness, the original individual is substituted by the new individual (line 23 and 24). This process is repeated for the same individual until improvements are no longer reached during an execution (line 29). Figure 7 uses the circle that represents the pupil border to illustrate this procedure.

Figure 7a shows a solution obtained by the genetic algorithm. Figure 7b shows the effect of shifting the center of the circle. After shifting the center, the radius is also ranged, as illustrated in Figure 7c. As there has been an enhancement in the fitness of that individual, the new individual substitutes the old one and another iteration of the local search algorithm is allowed. Once more, the amount of coincident pixels of each quadrant is evaluated, new coordinates for the center of the circle are generated (see Figure 7d) and new radiuses are investigated (see Figure 7e). The process is finished when no further enhancement in fitness is verified.



a) Solution found by the genetic evolutionary algorithm.

b) Applying local search: shifting the center of the circle.

c) Applying local search: ranging the radius.

d) Another iteration of the local search: shifting the center.

e) Subsequently ranging the radius.

Fig. 7. Illustration of the local search procedure.

The result of this process is such that the local search alters the individual and improves its fitness score. This individual returns to the evolutionary algorithm with a better fitness compared to the one it originally had, due to the cultural evolution obtained by the local search, which exploited the available knowledge of the problem.

3.4 Parameters definition

The implementation of a MA requires the definition of some variable parameters that are population size, number of generations, crossover and mutation rates.

The bigger the population size, more possible solutions are represented and higher the chance of good individuals to be present in the population. On the other hand, the increase of the population size results in the increase of the processing time.

If an evolutionary algorithm is well projected, one expects that, in a given generation, the individuals are more adapted or, at least, equally adapted to the environment than in the previous generations. Therefore, the bigger the total number of generations, the greater the convergence to a good solution, but it is evident that the processing time becomes longer. The number of generations can be the stop criterion.

The crossover rate represents the probability of a recombination of two selected individuals. If the crossover rate is 0%, the entire new generation is composed of exact copies of the parents. If it is 100%, all offspring is created by crossover.

The mutation rate, in this work, represents the probability of altering one or more genes of a chromosome. If the mutation rate is 0% nothing is changed, otherwise, if it is 100%, all individuals will suffer some alteration and the procedure will be similar to a random search.

In order to define a suitable set of parameters, the algorithm was applied numerous times to all images of the database and its behaviour was analyzed while the parameters varied. During the simulation, the population size ranged uniformly from 10 to 200 individuals. For a greater number of individuals, the processing time becomes inconveniently long. The crossover and mutation rates ranged uniformly from 0% to 100%. The time of convergence of the algorithm was the metric used to evaluate the parameters.

Best results were achieved using a population of 50 individuals, a crossover rate of 90% and a mutation rate of 20%. It was verified that, for the majority of images, the iris region was correctly segmented until the fifth generation. So, the number of generations was considered equal to 8, which was used as the stop criterion of the MA.

3.5 Experimental results

Systems based on any heuristic techniques, such as Memetic Algorithms, are non deterministic. It means that, a different solution can be obtained each time the system is simulated. After defining the parameters of the algorithm (population size = 50 individuals; crossover rate = 90%; mutation rate = 20%), the system was simulated many times. In each simulation, the algorithm was applied to all 1201 images and the number of images that were correctly segmented was summed. After all simulations, the arithmetic mean and the standard deviation of the results were calculated.

Considering all the simulations, the average number of images that were correctly segmented was 1166 from 1201 images. In an average of 13 images only the pupil border was correctly segmented and in an average of 12 images only the external border of the iris was correctly segmented. In an average of 10 images neither the pupil border nor the external border of the iris were correctly segmented. So the algorithm failed in segmenting a total of 35 images. It means that, when the algorithm is applied to images, an average 97.08% of the segmentations will be acceptable.

The standard deviation measured the spread of data about the mean and it was equal to 2.22. This is a small value and means that, for various simulations, the efficiency of the algorithm is always near the mean efficiency.

The results show that the method is robust enough to deal with not ideal conditions since a good efficiency rate could be achieved even with bad contrast and the presence of specular reflection in images.

4. Analysing the performance of iris segmentation methods applied to severely compressed images

An iris recognition system requires that the data related to the iris be stored for future comparison. The problem of storing the raw format of the image of the iris is that these images occupy a lot of space on disk and demand a high bandwidth for transmission. There is the alternative of storing only the code or template generated from the characteristics of the iris, which is considerably smaller. However, this code is generated by some proprietary algorithm, which would make the system dependent on some specific supplier. Presently, governmental organizations have demanded that biometric data be stored and recorded in the form of raw images, in order to have more negotiation freedom.

The fact that images are stored in raw format, that is, without any pre-processing, also has the advantage of allowing the data bank to take advantage of the inevitable evolutions of recognition algorithms in the future.

In this context, it is possible to see the necessity of image compression in biometry, and also the importance of knowing how much the captured image can be compressed without harming the performance of the iris recognition system.

Here, it will be verified what happens to the efficiency of the segmentation algorithms when the images undergo severe compression. For such, the fractal compression method that utilizes quadtree partitioning and the JPEG2000 compression are used with the objective of compressing the iris images at various compression rates, and analyzing up to what point compression would be viable.

Firstly, some features of the compression algorithms are presented and then, all the experimental results are shown.

4.1 Compression algorithms

Two different compression algorithms were applied to the iris images. One of them was the widely used JPEG2000 compression and the second one was chosen to be the Fractal compression that is based on different concepts.

The main characteristics of each of these techniques are presented in the following subsections which make clear that they have different principles. The main reason for the utilization of the second method was to guarantee that the conclusions about the behaviour of the segmentation algorithms are not dependent on the kind of compression technique that is applied.

4.1.1 JPEG2000 compression

JPEG2000 is an image coding system that was created by the Joint Photographic Experts Group committee in 2000. It is a more powerful version of JPEG coding offering improved image quality at very high compression ratios (Information Technology - JPEG2000 Image Coding System, 2004).

JPEG2000 has a superior compression performance over JPEG and it is attributed to the use of Discrete Wavelet Transform (DWT) and a more sophisticated entropy encoding scheme instead of the Discrete Cosine Transform (DCT). As a result, artefacts are less visible and there is almost no blocking. Moreover, JPEG2000 decomposes the image into a multiple resolution representation in the course of its compression process allowing local areas within each image tile to be encoded using different sub bands of coefficients (Christopoulos et al., 2000).

The advantage of the JPEG2000 over JPEG in terms of image quality is more evident at very low bit rates (Christopoulos et al., 2000) that correspond to severe compression, as studied here.

The JPEG2000 compression and decompression at various quality factors was performed using the Linux tools *pamtojpeg2k* and *jpeg2ktopam* from the JasPer JPEG2000 and Netpbm libraries.

4.1.2 Fractal compression

Fractal compression is a technique based on the principle that real images have self similarity and that these similarities consist of redundant information, which can be eliminated. For this, several transformations are applied over all or part of an image which in turn makes each part of the image to have a convergence to a common point called the attractor (Fisher, 1995). This way, the images can be stored through groups of affine transformation coefficients, consequently of reducing the memory utilized.

Fractal compression differs from other lossy compression methods, such as JPEG, in a number of ways. JPEG achieves compression by discarding image data that is not required for the human eye to perceive the image. The resulting data is then further compressed using a lossless method of compression. To achieve greater compression ratios, more image data must be discarded, resulting in a poorer quality image with a pixelized (blocky) appearance (Murray & Van Ryper, 1996). Fractal images are not based on a map of pixels. Once an image has been converted into fractal code its relationship to a specific resolution has been lost; it becomes resolution independent. The image can be recreated to fill any screen size without the introduction of image artefacts or loss of sharpness that occurs in pixel-based compression schemes.

Here, a fractal compression method with quadtree partitioning (Fisher, 1995) was implemented in Java environment. The implementation assures great processing speed of the images and wide parameter manipulation with the purpose of balancing image quality and compression rate. The coefficients of the compression algorithm were adjusted so as to obtain images with various degrees of compression.

4.2 Experimental results

The JPEG2000 compression and the fractal compression method were utilized to compress the images from the UBIRIS database (Proença & Alexandre, 2005) with the following compression rates (CR): 0.7, 0.5, 0.3 and 0.15. In this way it was possible to evaluate the efficiency of the segmentation methods when the images suffer compression from moderate rates (0.7 rate) to extremely severe compression rates (0.15 rate).

Table 1 shows the effect of compression on the images. In average, the original images have a size of 22000 bytes, consequently compression at rates of 0.7, 0.5, 0.3 and 0.15 produce compressed images with an average size of 15400, 11000, 6600, and 3300 bytes respectively. In relation to the original image, compression at these rates represents an average reduction factor of 1.5:1, 2:1, 3.5:1 and 6.7:1, respectively.

In order to apply both the Circular Hough transform and the proposed method based on memetic algorithms to localize the iris region in an image, it is necessary to generate an edge map from this image first, remembering that the Canny edge detector operator were applied and the same parameter values were used throughout the experimentation.

	Size	Reduction factor
Original	22 KB	1 : 1
CR = 0.7	15.4 KB	1.5 : 1
CR = 0.5	11 KB	2 : 1
CR = 0.3	6.6 KB	3.5 : 1
CR = 0.15	3.3 KB	6.7 : 1

Table 1. Effects of images compression - Average values.

Therefore before verifying the influence of compression in the iris segmentation, it is interesting to analyze the effect of compression on the edge map that is generated.

The edge map of each original image was compared to the edge map of the corresponding image compressed at the rates of: 0.7, 0.5, 0.3 and 0.15 using JPEG2000 compression and also Fractal compression. In average, 91.3% of the pixels considered to be edge pixels in the original image were also considered edge pixels in the edge map generated from the compressed image at a rate of 0.7 bpp when JPEG2000 compression was applied and 90.5% when Fractal compression was applied. For the edge maps generated from the images compressed at rates of 0.5, 0.3 and 0.15 bpp, this ratio was, in average of 81.1%, 78.5% and 49.8%, respectively, when JPEG2000 was used and 79.7%, 78.1% and 51.5%, respectively, when Fractal compression was used.

In the following subsections, the results obtained with each segmentation method are presented. It is important to emphasise that the segmentation rates that are presented in the following subsections represent the percentage of the images that were correctly segmented without compression which were also correctly segmented after compression. All results are summarized in Table 2.

4.2.1 Results of the circular Hough transform application

We observed that with the compression rate of 0.7 bpp, there was little variation in the edge map and as a result, from the original images that had their iris region correctly found (or segmented), the percentage of compressed images that had also the iris region correctly found was 99.8% for the JPEG2000 compression and 99.5% for the Fractal compression. This shows that moderate compression practically does not interfere in the efficiency of the Hough transform.

For the compression rates of 0.5 and 0.3, the edge map varied a little more, even so the algorithm was capable of correctly finding the iris region from, respectively, 99.4% and 98.4% of the images compressed by JPEG2000 and 99.3% and 98.5% of the images compressed by Fractal. This still represents a very acceptable efficiency.

With a compression rate of 0.15, the algorithm was capable of correctly finding the iris region from 93.5% of the images compressed by JPEG2000 and 94.7% of the images compressed by Fractal. The reduction of the efficiency of the algorithm was probably due to the more intense variation that occurred in the edge maps.

4.2.2 Results of the integro-differential operator application

When applying the integro-differential operator to original and compressed images it was observed that when the compression was performed with rates of 0.7 and 0.5 bpp, 100% of the original images that had their iris region correctly found were also properly segmented when compressed using both JPEG2000 and Fractal compression. Therefore, compression at these rates did not harm in any way the efficiency of the segmentation algorithm.

With a compression rate of 0.3 bpp, the algorithm was capable of correctly finding the iris region from 98.3% of the images compressed by JPEG2000 and 98.6% of the images compressed by Fractal, which represent a margin of error that can be ignored in face of the benefits of compression.

When a more severe compression rate was used (0.15), 88.2% of the images compressed by JPEG2000 and 89.7% of the images compressed by Fractal had their iris region correctly found. In this case, it is observed that the algorithm suffered greater interference from compression and for this technique this was especially due to the reduction in contrast between the white part of the eye and the iris and between the iris and the pupil in the compressed image.

4.2.3 Results of the eyelids and eyelashes detection method

The proposed method for the detection of eyelids and eyelashes was also applied to the original images and to compressed images with rates of 0.7, 0.5, 0.3 and 0.15 using JPEG2000 compression and Fractal compression. It was noticed that 100\% of the original images that had the eyelids and eyelashes correctly detected also had them correctly detected using any of the compression rates and any of the compressing algorithms. This shows that compression, even when severe, does not harm the algorithm utilized to detect eyelids and eyelashes.

4.2.4 Results of the proposed iris segmentation method

Applying the proposed method based on MA, 100% of the original images that had their iris region correctly found were also properly segmented when compression was performed with rates of 0.7 and 0.5 bpp using both JPEG2000 and Fractal compression. This shows that despite the little variation of the edge map the method was robust enough to guarantee the same efficiency with or without compression.

For the compression rates of 0.3 and 0.15 bbp, the algorithm was capable of correctly finding the iris region from, respectively, 98.9% and 95.2% of the images compressed by JPEG2000 and 99.1% and 95.7% of the images compressed by Fractal.

For a clearer comparison, the results are summarized in Table 2.

	CR	Circular Hough transform	Integro-differential operator	Eyelids and eyelashes detection	Proposed method
JPEG2000	0.7	99.8%	100%	100%	100%
	0.5	99.4%	100%	100%	100%
	0.3	98.4%	98.3%	100%	98.9%
	0.15	93.5%	88.2%	100%	95.2%
Fractal	0.7	93.5%	100%	100%	100%
	0.5	99.3%	100%	100%	100%
	0.3	98.5%	98.6%	100%	99.1%
	0.15	94.7%	89.7%	100%	95.7%

Table 2. Effects of images compression in the segmentation stage. The values correspond to the percentage of the original images that were correctly segmented without compression which were also correctly segmented after compression.

5. Conclusion

The segmentation of the iris region is the first processing stage of an iris recognition system. The efficiency of this stage is essential to the success of the recognition task. In this work some traditional iris segmentation methods were firstly presented and evaluated by applying them to images from the UBIRIS database. Those images were chosen once they were captured under not ideal conditions and so, were challenging enough to the segmentation stage, what allows knowing how the methods would perform in real situations.

Then, a new segmentation method based on Memetic Algorithm that is an evolutionary algorithm was proposed and evaluated. Firstly, the basic concepts of Memetic Algorithms were described and then, the implementation of the proposed method was carefully detailed. The experimental results indicate that the proposed method is efficient at localizing the iris region in an image of an eye. The innovation related to the use of evolutionary algorithms to localize the iris region can be further explored in order to achieve even better results.

This work also examined the influence of severe image compression on the segmentation stage. Images of the UBIRIS database were compressed using two different compression algorithms: JPEG2000 and Fractal compression. It was possible to conclude that, in general, the most traditional boundary-based and template-based algorithms for iris segmentation continue presenting good performance even with compressed images.

Finally, it is concluded that in a complete iris recognition system, the segmentation stage will not represent a bottleneck, that is, it will not compromise the performance of the system in case compressed images are used.

The next work to be suggested is to perform a research of how compression interferes in the other processing stages, especially in the system's ability to recognize individuals precisely.

6. References

Canny, J. (1986). A computational approach to edge detection, *IEEE Transactions on PAMI-8*, No.6.

Christopoulos, C. ; Skodras, A. & Ebrahimi, T. (2000). The JPEG2000 still image coding system: An overview, *IEEE Trans. Consum. Electron*, Vol. 46, No.4, pp. 1103-1127.

Daugman, J. D. (1993). High confidence visual recognition of person by a test of statistical independence, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.15, No.11, pp. 1148-1161.

Fisher, Y. (1995). Fractal Image Compression - Theory and Application*, New York: Springer- Verlag*, 1995. 341 p. ISBN: 0-387-94211-4 (New York) - 3-540-94211-4 (Berlin)

Gonzalez, R. C. & Woods, R. E. (2002). *Digital image processing* (2nd edition), Prentice Hall, Upper Saddle River

Huang, J.; Wang, Y. ; Tan, T. & Cui, J. (2004). A new iris segmentation method for recognition, *In: Proceedings of the 17th International Conference on Pattern Recognition*, Vol.3, pp. 554- 557, DOI: 10.1109/ICPR.2004.1334589.

Information Technology - JPEG2000 Image Coding System (2004). Int. Std. ISO/IEC 15444-1, 19 March 2011, Available from: http://www.itu.int/rec/T-REC-T/en.

Ma, L.; Wang Y. & Zhang, D. (2004). Efficient iris recognition by characterizing key local variation, *In: IEEE Transactions on Image Processing*, Vol.13, No.6, pp 739-750.

Masek L. (2003). *Recognition of human iris patterns for biometric identification*, Master Thesis. School of Computer Science and Software Engineering, University of Western Australia.

Moscato P. (2001). *NP Optimization Problems, Approximability and Evolutionary Computation: From Practice to Theory*, Ph.D. dissertation, University of Campinas, Brazil.

Murray, J. D. & Van Ryper, W. (1996). *Encyclopedia of Graphics File Formats*, (2nd edition), ISBN: 1-56592-161-5.

Proença, H. & Alexandre, L. A. (2005) UBIRIS: a noisy iris image database, ICIAP 2005, *13th Int. Conf. on Image Analysis and Processing*, Cagliari, Italy, 6-8 September 2005, Lect. Notes Comput. Sci., 3617, pp. 970-977, ISBN 3-540-28869-4. http//iris.di.ubi.pt

Wildes, R. P. (1997). Iris recognition: An emerging biometric technology, *Proceedings of the IEEE*, Vol.85, No.9, pp. 1348-1363.

Biometric Systems, Design and Applications
Edited by Mr Zahid Riaz

ISBN 978-953-307-542-6
Hard cover, 262 pages
Publisher InTech
Published online 21, October, 2011
Published in print edition October, 2011

Biometric authentication has been widely used for access control and security systems over the past few years. The purpose of this book is to provide the readers with life cycle of different biometric authentication systems from their design and development to qualification and final application. The major systems discussed in this book include fingerprint identification, face recognition, iris segmentation and classification, signature verification and other miscellaneous systems which describe management policies of biometrics, reliability measures, pressure based typing and signature verification, bio-chemical systems and behavioral characteristics. In summary, this book provides the students and the researchers with different approaches to develop biometric authentication systems and at the same time includes state-of-the-art approaches in their design and development. The approaches have been thoroughly tested on standard databases and in real world applications.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:

Milena Bueno Pereira Carneiro, Antônio Cláudio P. Veiga, Edna Lúcia Flôres and Gilberto A. Carrijo (2011). Solutions for Iris Segmentation, Biometric Systems, Design and Applications, Mr Zahid Riaz (Ed.), ISBN: 978-953-307-542-6, InTech, Available from: http://www.intechopen.com/books/biometric-systems-design-and-applications/solutions-for-iris-segmentation

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821