Creating an Iris Image from a given Iris Template

A. de Santos, C. Sánchez, and V. Jara

***Abstract*—An Iris Detection System is known to be one of the strongest systems in terms of security. One of the main security aspects of this system relays on the incapability to reconstruct the original iris image from the iris template, i.e. that binary string able to provide the enough information to identify and/or authenticate a certain user. However, this paper proposes a method in order to carry out the solution for such a problem. The algorithm is based on evolutionary strategies, and intends to find an image whose iris template attends to be so similar as required to a given iris template. Results will highlight how this algorithm achieves the required aim, and how the performance can be as accurate as desired.**

***Index Terms*—Authentication, Biometry, Genetic algorithms, Evolutionary Strategies, Identification, Iris Detection System, Iris falsification, Iris Template.**

I. Introduction

SINCE an Iris image is different even between twins [6], [7], [9], the idea of replicating an Iris seems to be a very difficult task. Actually, falsifying any Biometric template (fingerprint, face, Iris, DNA,...) is a hard goal, [17]. In fact, current Biometric Systems based on Iris detection relay one of their strengths on the impossibility to falsify an Iris image from its template [7], [9], [10], [18] .

However, based on [4], a very interesting solution for a similar problem is provided in the field of a Fingerprint-based Biometric System. Could it be possible to do something similar for an Iris System?

No solutions have been provided yet, since tackling the problem of falsifying an Iris Image can require a very high effort regarding the tissues of the Iris itself, and its anatomy, much more complex than a fingerprint, [7].

Furthermore, the Iris template is so highly related to the scheme of extracting the characteristics from the users, that such an algorithm, able to break the system down, should be also strongly tied to such a specific scheme, [5], [7], [10], [18]. This possible schemes will be described in Appendix A.

On the contrary, this algorithm will provide a general solution for most of the Biometric Systems based on Iris detection due to the fact that all of them extract a similar template from the Iris image. Moreover, it can be said that those extraction algorithms differ only in the number of points of the Iris template, and which regions within the Iris are read to obtain the Iris template. The procedure of this algorithm can be easily extended to each of those Biometric Systems. Section II-B will introduced how this common template for all Biometric Systems based on Iris Detection is implemented.

Evolutionary Computing, [11], [12], is the method purposed to solve this problem. This research area of computer science provides important tools to evolve raw possible solutions till they achieve a required aim or goal. Actually, they are used to solve Optimization Problems, [8], [14], [15].

As a future work, and beyond of the scope of this algorithm, an inverse system could be implemented from the binary data provided by a real system, for instance, those presented in Section A, so that, the row Iris template could be obtained.

II. From Template to the Image

ONCE the problem has been introduced, a wider description of the problem is necessary to tackle the problem properly, Section II-A. Feature extraction will be explained, Section II-B in order to understand how the whole problem can be simplified. Since this approach needs Evolutionary Strategies [19], Genetic Algorithms [8], [14], [15], and Evolutionary Computing [2], [3], [16], tools, these are presented in Section II-C together with a detailed explanation of the constitution of the algorithm itself. Therefore, a brief section comes up to provide visual results to what the previous algorithm has achieved. This will be shown in Section II-D. Finally, quality and temporal results are provided in Section II-E.

A. Problem Statement

WITHIN this section, the algorithm is briefly described. The aim of this section consists only of presenting a general overview of the whole process and how the algorithm fulfil its purpose.

The algorithm will take as input data an Iris template from a determined user. This user will be referred as DU and will access the system by an image of one of its Irides. How this template has been obtained in a real system is beyond of the spot of this algorithm, however, in Appendix A will be indicated how to capture such data for each system.

The Iris template which belongs to DU, (Determined user), will be referred as T_{DU}. From that template T_{DU}, the algorithm, referred from now as $\mathcal{A}$, will create an Image whose template is so similar to T_{DU} as desired.

Let FU be the reference that stands for False User (again, FU will be represented by an image of an Iris), who actually intends to be DU. In order to achieve this purpose, FU needs of $\mathcal{A}$ the information obtained from the system, i.e. T_{DU}. If this goal is achieved, FU will obtain an image whose template (T_{FU}) will be considered as T_{DU} in different Biometric Systems.

Mathematically, $\mathcal{A}$ can be considered as a function with a template as its input, and an image as its ouput. This can be expressed as in Eq. 1.

A. de Santos, C. Sánchez and V. Jara are in Grupo de Biometría y Tratamiento Numérico de la Información.

A. de Santos (alberto@cedint.upm.es) and C. Sánchez (csa@mat.upm.es) are in Centro de Domótica Integral (CeDInt).

V. Jara (vjara@mat.upm.es) is in Dpto. de Matemática Aplicada a las Tecnologías de la Información.

ETSI de Telecomunicación. Universidad Politécnica de Madrid.
Ciudad Universitaria s/n, 28040 Madrid (Spain)

$$\begin{array}{rrr} \mathcal{A}: & \mathcal{T} \longmapsto & \mathcal{D} \\ & T \in \mathcal{T} \longmapsto & \mathcal{A}(T) \in \mathcal{D} \end{array} \tag{1}$$

where $\mathcal{T}$ represents the set of all possible templates for a given Biometric System. For the sake of simplicity, $\mathcal{T}$ will gathered elements of 256 points of length, with values belonging to the set $[0, 255]$ according to the greyscale representation of an Image, [13]. In Appendix A, several $\mathcal{T}$ will be presented in relation to each Biometric System.

Furthermore, $\mathcal{D}$ represents the set of all possible Images of a given database. Again, for the sake of simplicity, this algorithm has been implemented considering only images which belong to CASIA v3 database, [20]. Although each Biometric System is strongly related to a given database, the main idea of the algorithm $\mathcal{A}$ can be easily extended to other databases, since the $\mathcal{A}$ hardly depends on the image representation.

Moreover, and continuing with the mathematical representation of the problem, let $\mathcal{Z}$ be the algorithm which carries out the extraction of the template from a given image, which will be extensively defined in Section II-B, and let $\eta \in \mathbb{R}$ be the degree to what extend two templates are similar or not. In a binary representation, $\eta \in \mathbb{R} \cap [0, 1]$, by averaging the result with the length of the binary data. In these terms, η is defined as follows in Equation 2:

$$\begin{array}{rrr} \eta: & \mathcal{T} \times \mathcal{T} \longmapsto & \mathbb{R} \\ & T_1, T_2 \in \mathcal{T} \longmapsto & \eta(T_1, T_2) \end{array} \tag{2}$$

and where η can be implemented by a wide range of operators, withouth loss of generality. However, specifically for this implementation, Hamming distance between two vectors will work as η function, and furthermore, the lower η is, the more similar T_1 and T_2 are.

Finally, gathering Eq. 1 and 2, the problem can be stated mathematically as follows:

Given T_{DU} and η_0, implement $\mathcal{A}$ in order to verify that $\eta(\widetilde{T}, T_{DU}) \leq \eta_0$, where $\widetilde{T} = \mathcal{Z}(\mathcal{A}(T_{DU}))$

Having the problem statement already defined, it remains to describe how $\mathcal{Z}$ is implemented specifically for this procedure, Section II-B, and how $\mathcal{A}$ is developed in terms of Evolutionary Strategies, Section II-C.

B. Feature Extraction

FEATURE Extraction represents the main step in a Biometric System. This extraction begins after pre-processing, segmentating the Iris image and identifying its different parts like Pupil, Iris, Eyelids and so forth, [9]. During the process of Iris Detection, two kind of templates must be distinguished: one directly extracted from the Iris which is considered as a 'raw' Iris template, and the final template as a result of a certain processing, clearly determined by the different schemes used in each Biometric System. This schemes are presented in Appendix A.

In this algorithm, only 'raw' Iris template, as described previously, is used, since this template is common to most of the Biometric System based on Iris Detection, despite of the difference regarding the number of points of such a template, and the regions from which the features are extracted. Such 'raw' Iris template will be referred only as template, for the sake of simplicity in the language.

As introduced in Section II-A, $\mathcal{Z}$ will carry out the extraction of the features from the image. In order to fulfil this aim, a circular crown will be the tool to read the template from the Iris. This circular crown is obtained by two concentric circles, whose center depends on the Biometric System (Iris Center or Pupil Center). In this implementation, the center of the circular crown coincides with the center of the pupil.

The circular crown barries 256^{o}, 360^{o} in other Biometric Systems as it can be seen in Appendix A, symmetrically distributed with the vertical axis which goes through the previous selected center. In case of a circular shift of the Iris image, the previous vertical axis must be shifted in the same proportion. However, as it can be seen in Section II-D, this is not needed for the implementation of $\mathcal{Z}$, since the image used to be falsified is fixed, and the shifted angle is already known.

Evidently, the main parameter of the circular crown is the radius, in other words, the width of the crown. To extract then the template, for each angle, values belonging to the straight line which joins both points are averaged, obtaining one point for each angle, i.e., 256 points. It must be considered the discrete nature of the representation offered by any image, independently from the selected database. Mathematically, given an image $I \in \mathcal{D}$, the template $T \in \mathcal{T}$ is obtained as follows in Eq. 3

$$T = \mathcal{Z}(I) \tag{3}$$

Let X and Y be the set of points in the horizontal and vertical axis respectively barried by each straight line when joining the points of the two concentric circles for each given angle. In mathematical terms, $C_1(\alpha)$ and $C_2(\alpha)$, represent the two points of the concentric circles at an angle $\alpha \in \mathbb{N} \cap \ [1, 256]$. The straight line, s, which will join $C_1(\alpha)$ and $C_2(\alpha)$, will store which horizontal and vertical pixel coordinates are 'touched' by s, in X and Y, respectively.

After finding out the sets X and Y, those values read when procesing the template, are stored in a matrix V. Section II-D must be referred here, because there, it is shown what this matrix V is used for. Also, it must be pointed out that this matrix V is only found out once, in other words, when extracting the template from the original image, V is not obtained. V is calculated once, for that image used to make the falsification, as it can be seen in Section II-D.

However, and despite of not being always calculated, V is related to T, the template obtained from the image. The relation is as follows in Eq. 4, for a given $i \in \mathbb{N} \cap [1, 256]$

$$T(i) = \frac{\sum\limits_{j} V(i,j)}{L_i} \tag{4}$$

where L_i is the length of the rows of V. It is clear that V required 256 rows, the same number as values the template of an image requires.

Finally, a conclusion is followed from the previous definitions. If $I \in \mathcal{D}$ is considered as a set of points, then V is the subset of I whose points are used by $\mathcal{Z}$ to build the template T. This introduces an important idea which will be the base for the next algorithm, actually the core of the algorithm $\mathcal{A}$.

C. An evolutionary approach

PREVIOUS section introduced an important idea, which is developed here. An image $I \in \mathcal{D}$ can be used as a 'background' image, keeping the characteristics of the pupil, eyelid, skin, and so forth, since this elements are considered in the Iris Detection preprocessing but are not checked whether they belongs to the user who pretends to access the system. Then changing those values in V, located within the image I by X and Y, a new template can be obtained in terms of V. Due to the fact that $T \in \mathcal{T}$ depends only on V, how to alterate those values in V becomes the current problem to be solved.

Here comes Genetic Algorithms and Evolutionary Strategies to offer a possible solution for the previous problem since it can be seen as a minimization/maximization problem, in other words an optimization problem.

Mathematically, the problem can be stated as follows in a similar way as it was defined in Section II-A.

Given $\underset{\sim}{T_{DU}}$*, minimize* $\eta(T_{DU}, \widetilde{T})$ *under a threshold* η_0*, where* $\widetilde{T} = \mathcal{Z}(\mathcal{A}(T_{DU}))$

In the following subsections, the problem will be exposed in terms of 'evolutionary strategies'.

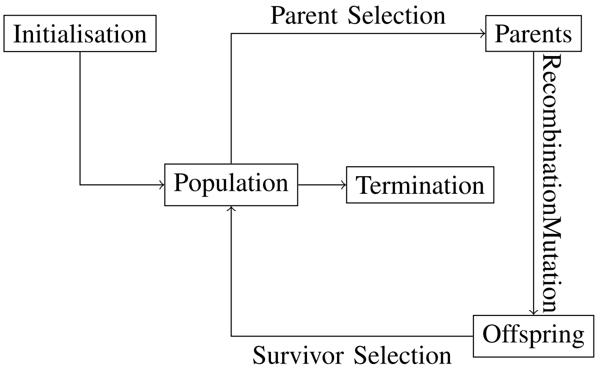
1) An introduction to Evolutionary Computing: Evolutionary Computing is inspired in process of natural evolution, and nowadays is one of the main research areas within computer science. Since a wide explanation of Evolutionary Computing [1],[12] is far beyond of the scope of this paper, this section will only provide the reader with the basic knowledge to understand the algorithm $\mathcal{A}$.

An evolutionary algorithm (EA) is a very suitable procedure for tackling problems of minimization and maximization. As an overview, a pseudocode is presented with the main parts of a evolutionary algorithm (EA). All these parts will be extended in the subsequent subsections.

Algorithm 1 Pseudocode of an EA

```
BEGIN
  INITIALISE population with random candidate solutions;
  EVALUATE each candidate;
  REPEAT
    SELECT parents;
    RECOMBINE pairs of parents;
    MUTATE the resulting offspring;
    EVALUATE new candidates;
    SELECT individuals for the next generation;
  UNTIL TERMINATION CONDITION is satisfied
END
```

A schematic view of an evolutionary algorithm (EA) is shown as follows:



Before extending all the components of the previous algorithm, a suitable representation, defined as *genotype* [12], for the problem, defined as *phenotype* [12], must be selected. As it was said in Section II-A, each template $T \in \mathcal{T}$ verifies to be a vector of 256 components, with each component belonging to the interval $\mathbb{N} \cap [0, 255]$. The idea consists of 'evolving' each component of the previous template T, in other words, split the problem into 256 independent problems.

However, a consequence of this consideration could be an increment in time of the whole algorithm, but this is not important since time in 'hacking' activities is not a limiting factor and, as it will be seen in Section II-E, the EA is not very time-consuming despite of this fact.

So, the representation of the i-component of T will correspond to the average of the components within the row i in the matrix V. Representation for the subproblem i will be then a vector called $v_i =< a_1, a_2, \ldots, a_n >$ with n components.

2) Fitness function: This funtion will provide the algorithm with enough information to continue or stop with the procedure, since *TERMINATION CONDITION* will be given by that function. There exists several options for the selection of an evaluation function, [12]. Considering the fact that for the subproblem s, the precise result would be

$$|T(s) - \overline{v}_s| = \left| T(s) - \frac{\sum_{i=1}^{n} a_i}{n} \right| = 0$$

then the fitness function will be $|T(s) - \overline{v}_s|$.

3) Population and initialisation: The population of each subproblem will consists of 50 members, each of them different to each other, in order to ensure the property of *diversity*. Each member of the population will be created from the image considered as a 'background', i.e., from matrix V. Let consider again the subproblem s, with $s \in \mathbb{N} \cap [1, 256]$. The row s of the matrix V is stored in Σ. After that, two values are extracted from that vector Σ: maximum, maxΣ, and minimum, minΣ, value. From those values, a uniform distribution is created, choosing randomly, and based in the previous distribution, the values of the 50 members of the population.

When the population of the subproblem s has been created, they present a presolution quite close to the final result which still keeps unknown. This is one of the reasons why this EA is very fast.

4) Parent Selection: Parent Selection will provide the EA with those components of the population whose results have been closer to the required fitness function. In the nature, those individuals would have more opportunity to mate and form a new offspring. This part deals with this fact, and intends to simulate such an event. However, not only must those most suitable be selected, but those also whose fitness function result has not been so precise. Otherwise, the EA will come up with an incorrect solution.

For that, and considering that each component of the population of the subproblem s will provide an error given by $|v_s - \overline{v}_i| = \xi_i$, then an error distribution is created based on ξ. Then, those members of the population whose error is closer to 0 will gather more probability to be chosen, allowing those with a bigger error to be selected too, but with less chance to be chosen.

Since this problem has only one minimum, then parent selection can be not so strict. A multi-minimum/maximum problem will require a more elaborated parent selection. However, and considering the constraints of this problem, this parent solution will provide with a good and fast selection.

In the literature, this Parent Selection is similar to Fitness Proportional Selection.

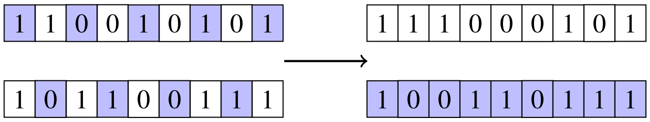
5) Recombination: Once the parent selection has been carried out, the new offspring will be created. Several recombination operators exist within the literature, but however, and considering the properties of each member of the population two operators are considered Whole Arithmetic Recombination and a Crossover operator with probability p_c.

The Whole Arithmetic Recombination takes two parents, v_n and v_m, and computes the next operation:

$$Child_1 = \alpha \cdot v_n + (1 - \alpha) \cdot v_m$$
$$Child_2 = \alpha \cdot v_m + (1 - \alpha) \cdot v_n$$

In other words, this operator works by taking the weighted sum of the two parental alleles for each gene. It means each child contains more information of a different parental allel, obtaining an offspring more able to survive in subsequent performances, [11]. A modification must be done considering the nature of each allele. Since the values in each allele are in the set $\mathbb{N} \cap [0, 255]$, then the offspring must verify the same condition. So that, the previous children are rounded to the closest integer value. Furthermore, $\alpha = .3$ in this EA, considering the fact that $0 < \alpha < 1$, as it is suggested in [12].

Due to the sctructure of the individuals, creating a crossover operator is a very easy task. Despite of the amount of different operators, only a pre-established crossover is carried out. Have a look to the next picture.



1 1 0 0 1 0 1 0 1 → 1 1 1 0 0 0 1 0 1

1 0 1 1 0 0 1 1 1 → 1 0 0 1 1 0 1 1 1

On the left, parents are presented. Only binary notation is used, for the sake of simplicity. On the right, childs are presented. Furthermore, crossover operator is performed with a probability $p_c = .7$. Lower values of this probability will make the EA not to converge in a reasonable time, [12].

6) Mutation: As Whole Arithmetic Recombination is always performed and Crossover operator is carried out only with a probability of p_c, Mutation is performed with a probability p_m.

Mutation is an operator which modifies the offspring by changing several (from none to all) elements within the alleles. In this sense, mutation can create a very outstanding individual from a very poor allele or viceversa. Despite of many authors within the literature, Mutation cannot be rejected so easily. It is a very powerful operator, although it looks very simple.

In this EA, a uniform mutation operator is carried out with a distribution made similarly to that from the initialisation. Firstly, which values of the individuals are to be changed are selected randomly. After this, a distribution is made based on its values, creating then a new allele with the previous selected values changed according with the distribution.

The mutation probability is $p_m = .25$. Higher values will not make the EA converge, [12].

As an overview of Recombination and Mutation, Table I shows a resume of the main parameters of each operator:

TABLE I
PARAMETERS VALUES OF THE DIFFERENT OPERATORS

Operator	Parameter	Value
Whole Arithmetic Recombination	α	.3
Crossover Recombination	p_c	.7
Mutation	p_m	.25

7) Survivor Selection Mechanism: This mechanism is responsible for managing the process whereby the population of parents (in the literature, μ) and the new offspring (λ) is reduced to the size of the population. There are two kind of Evolutionary Strategies: (μ, λ) and $(\mu + \lambda)$. The former selects μ individuals from λ individuals of the offspring. The latter selects μ individuals from $\lambda + \mu$ individuals of the population formed by parents and children. The former, i.e., (μ, λ) is used for solving very complex functions. This strategy can achieve the self-adaptation of mutation rates.

For this EA, Elitism [12] has been chosen as the most suitable mechanism of survival selection. Elitism combines Age-based replacement (those individuals whose time within the process is very long are excluded) and Fitness-based Replacement, in the same way parent selection is carried out.

With all this subsection, EA has being extensively explained. Implementation of this algorithm is shown in Section II-D and its results can be shown in Section II-E.

D. Implementation

THIS section will gather all the previous sections providing visual examples in order to establish all the previous definitions, and mainly a final result of the genetic algorithm. This section is included only to assemble the previous knowledge. Next section will provide the reader with results about the performance of the algorithm and a more precise form of measuring how the whole algorithm meets its goals and tackles the problem.

First of all, the determined user (DU) presented in Section II-A access the system with the image shown in Fig. 1

The template of the previous user, T_{DU} is shown in Fig. 2.

From now till the end, the EA does not know anything about the image represented in Fig. 1, but only its template.

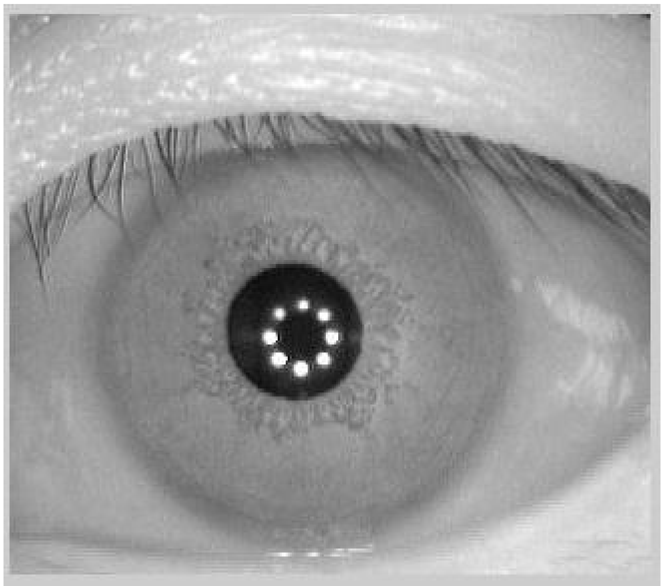


Fig. 1. Iris Image of DU

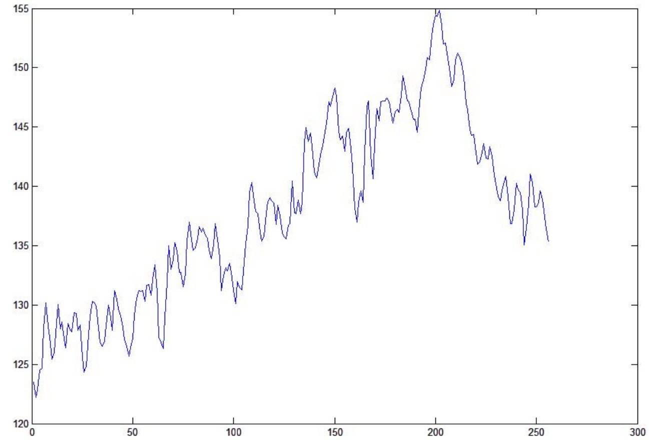


Fig. 2. T_{DU}, Iris Template of DU

Taken as an image background, considering the advantages of the established properties of the image, matrix V, X and Y are extracted, as seen in Section II-B. Next picture is presented in Fig. 3.

This image was selected as the background because the deviation due to a circular shift of the iris is almost zero. However, each image can be selected as a background image, but matrices V, X and Y must be recalculated. This image will be the image that a False User (FU) would present.

Finally, after performing the previous EA, the FU has it purpose fulfill: An image, whose template $T_{FU} = \widetilde{T}$ is so similar to T_{DU} as desired. As a final detail, since the color properties of the different templates are obviously different too, an offset is carried out in order to camouflage the section of the 'background' image which has been evolved, and the original part of the image. The final image, shown in Fig. 4, does not seem to be very human, however, the system cannot distinguise between Fig. 1 and Fig. 4.

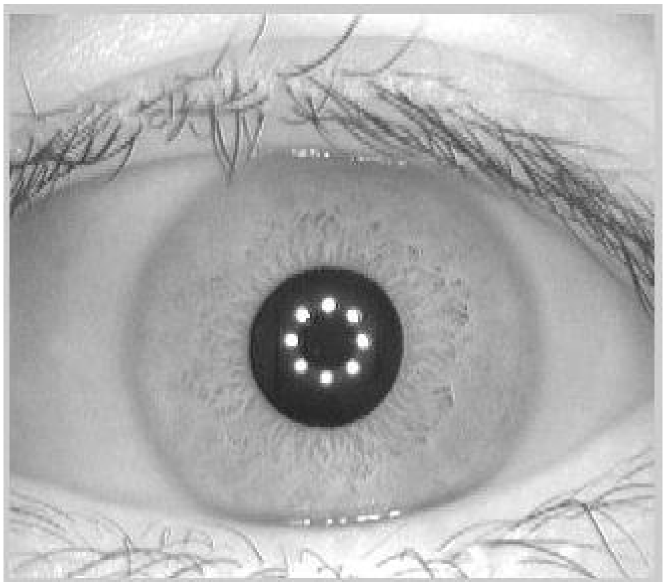


Fig. 3. Background Image

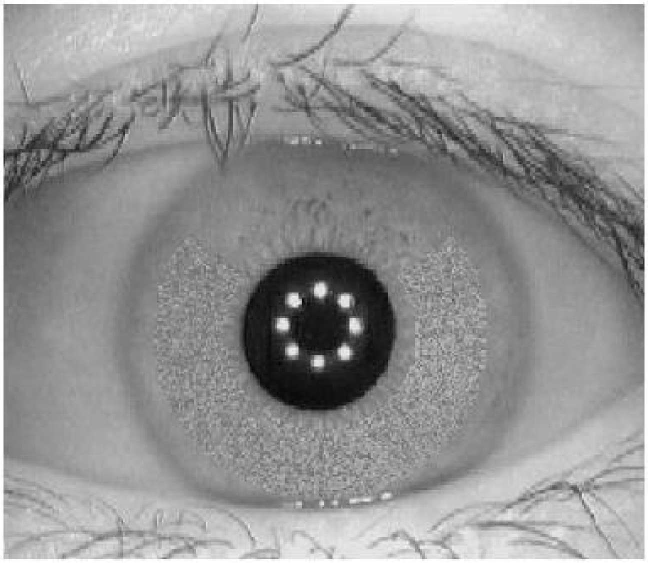


Fig. 4. FU, false Image

E. Results

THE aim of this section is twofold: firstly, a quality measure of the performance of the algorithm $\mathcal{A}$ is carried out, secondly, a time study is also considered, althought as it was said before, time is not an important factor in 'hacking' activities.

1) Quality Measure: Within this section, the performance of the algorithm is measured. Two points of view will be considered regarding this aim. First of all, a visual comparison between different performances of the algorithm for a given user DU is carried out. This comparison is shown in Fig. 5. For the sake of clarity, a darker line is shown to represent the template captured to DU, in other words T_{DU}. The rest of the lines, those less dark, stand for several performance of the algorithm $\mathcal{A}$, where η_0 is set to 3, i.e. the EA will stop when

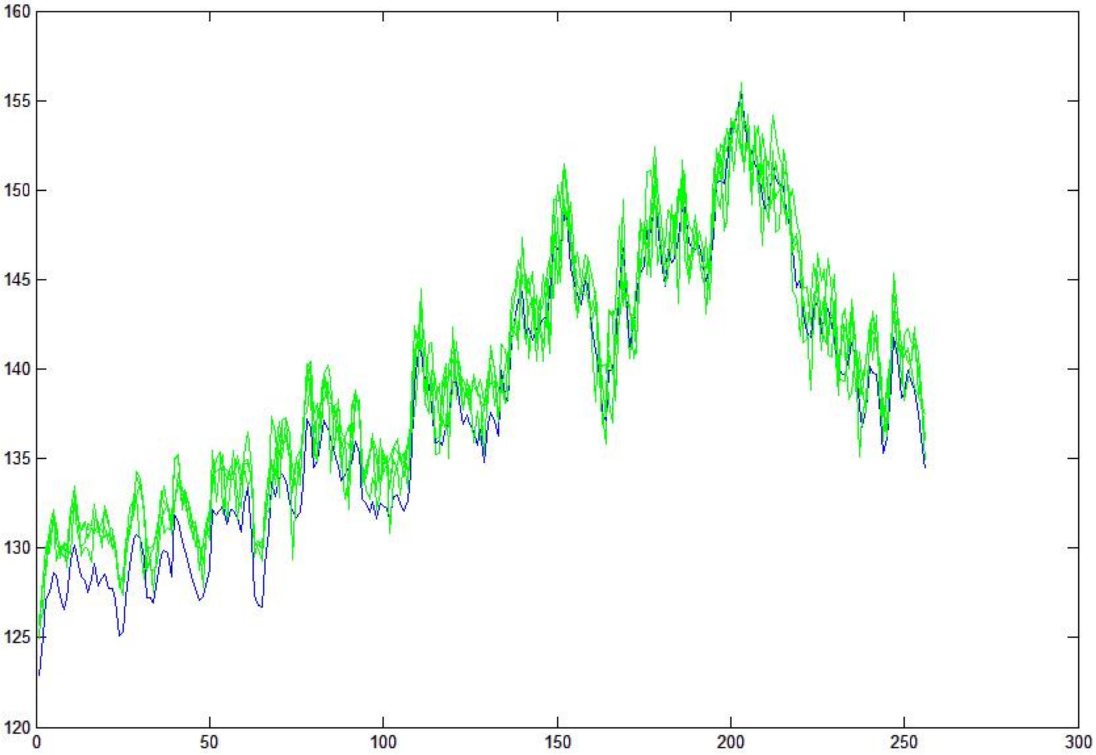


Fig. 5. Several performances of Algorithm $\mathcal{A}$

each point of the new template $\widetilde{T}$ is at least 3 points of value up or down in terms of η, as in Section II-A was established.

Probably, an inexperienced (in terms of Biometry) reader could think that $\widetilde{T}$ does not fit exactly T_{DU} concluding that the result is not to be good. However, as it is shown in Fig. 6, even being the same user does not imply that the template T_{DU} could keep invariant to different accesses. Actually, this is one of the main problems Iris Detection Systems tackle with.

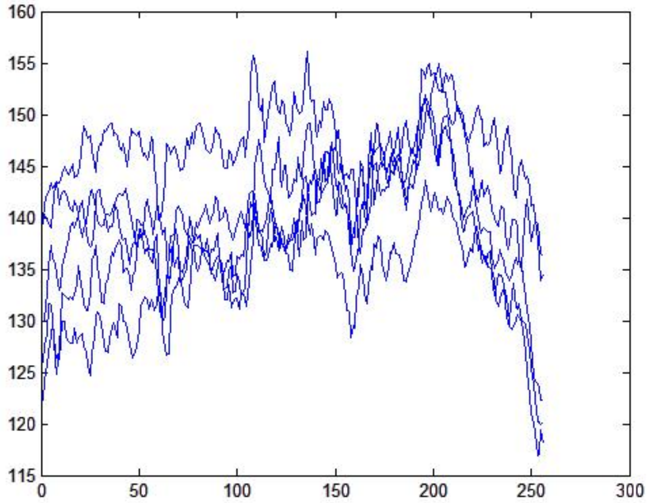


Fig. 6. T_{DU} for different images of a same user. Note the variability of T_{DU} for each access.

So, as it can be observed in Fig. 5, the algorithm $\mathcal{A}$ provides of a very good approximation to T_{DU}. Actually, this approximation is so accurate, that posterior signal processing operations [13] must be carried out in order to make $\widetilde{T}$ not so similar to T_{DU}. The Iris Detection System could even suspect of such a good replication of a user.

Considering the naïve similarity function, η, between two templates belonging to $\mathcal{T}$, Table II is shown, gathering the scoring, in terms of η of the different performances.

TABLE II
NUMERICAL RESULTS PROVIDED BY η

$\mathcal{A}$	Same User
491.8	2813.9
507.6	2106.5
529.4	917.5
502.4	1628.3
497.2	1451.9

In the first column, the different values of η for repeated performance of $\mathcal{A}$ are provided. On the other hand, in the second column, the different values of η for repeated access of the same user is also provided. This table comes up to result the differences, in terms of η, in both situations, where despite of the naïve approximation of η, scores in the former column are lower than in the latter column, showing up that the result of the performance of $\mathcal{A}$ is even better than another image of the same user.

2) Temporal Measure: In this section, a brief temporal study is carried out, since this factor is not a constraint factor for the algorithm. Although the Evolutionary Algorithms are supposed to be very time-consuming, this algorithm $\mathcal{A}$ is quite fast compared to others. In average, the time is $\bar{t} = 7.25s$, a very fast algorithm considering its evolutionary origins. This algorithm has been performed in a Pentium IV, with 2GB of RAM.

III. Conclusion

As a final overview, the algorithm $\mathcal{A}$ provides a very accurate solution for the stated problem. Future work must be done in the final aspect of the obtained image, since it does not seem to be a human eye for a human person, despite of being unrecognizable for a computer. Furthermore, the algorithm can be improved by beginning from a binary template, however this will constraint the algorithm to a certain scheme, and what is more, the algorithm can be tested in other databases different from CASIA.

The reader can easily imagine 'obscure' applications due to the evil behaviour of this algorithm, however, this algorithm can improve current biometric systems, making them stronger to these kind of attacks and preventing them against image injections, and protecting or masking somehow the template extracted from the user.

Appendix A
Schemes of different Biometric Systems to process Iris templates

SEVERAL schemes have been proposed within the literature, however, only three of them are briefly resumed here. The main purpose of this Appendix regards how the templates extracted from the Iris image are related to the Iris Detection System, and how the previous algorithm $\mathcal{A}$ could fit in such schemes.

Firstly, the scheme proposed in [6] extracts the features in a similar way to what has been exposed here, however, the image preprocessed, with the isolated Iris, is processed now by a Gabor filter using different areas read from the previous Iris. In this case, the algorithm $\mathcal{A}$ should be changed drastically because the feature extraction is quite different.

Secondly, the scheme proposed in [18] considers as a template only two sections of the whole iris, so the template is changed a bit, but however, this fact does not require a great effort for the algorithm $\mathcal{A}$ to find the solution. Only matrix V, X and Y, should be reimplemented according to the section of the Iris from which the template is extracted out. As in the previous scheme, a scheme based in Gabor filters is carried out, codifying in the same way the different responses of such a filter.

Finally, the scheme proposed in [10] considers a different template from the typical different crown. In this paper, the template is extracted based on two triangles concentric with the pupil. After that, the intersection of the previous triangles is considered, obtaining then six exterior triangles. For each triangle, its three vertexes are considered. From each vertix, a straight line between this vertix and its opposite side is obtained iterately with increments of angle of α. This straight line is similar to s in Section II-B, and actually the values 'touched' by the previous straight line are averaged and gathered in the final template for each vertix of each triangle.

This allows the system to select how many points could form the template. A fine explanation of this scheme is out of the aim of this Section, but a knowledge about the template is necessary to understand that this scheme is more difficult to be 'cracked' since the length of the templates can be highly increased and, which is more, the order to obtain the template, regarding vertixes and triangles, can be easily changed making almost impossible to reconstract the Iris image from the template. Furthermore, the template is changed by means of Wavelets transforms and Zero-cross processing, obtaining finally a binary data.

This is then an example, where the template has been improved for the sake of a greater security within the system.

So, as an overview, it is clear now how the template is strongly determined by the Biometric System. This is the reason why this paper intends to be so general, because trying to solve a specific case will mean to forget about the other cases.

Acknowledgment

The authors would like to thank Proyecto CENIT Segur@: Seguridad y Confianza en la Sociedad de la Información, financed by Ministerio de Industria, Turismo y Comercio.

References

[1] T. Bäck, *Evolutionary Algorithms in Theory and Practice*, Oxford University Press, Oxford, UK, 1996.

[2] T. Bäck, D. B. Fogel, Z. Michalewicz, Eds. *Evolutionary Computation 1: Basic Algorithms and Operators.*, Institute of Physics Publishing, Bristol, 2000.

[3] T. Bäck, D. B. Fogel, Z. Michalewicz, Eds. *Evolutionary Computation 2: Advanced Algorithms and Operators.*, Institute of Physics Publishing, Bristol, 2000.

[4] R. Capelli, A. Lumini, D. Maio and D. Maltoni, *'Can Fingerprints be reconstructed from ISO Templates?'*, in Proc. *International Conference on Control, Automation, Robotics and Vision (ICARCV2006)*, Singapure, December 2006.

[5] C. N. Chun, R. Chung, *Iris Recognition for Palm-Top Application*, First International Conference, ICBA 2004, Hong Kong, LNCS 3072, pp.426-433, 2004. Springer-Verlag Berlin Heidelberg.

[6] J. Daugman, *High Confidence Visual Recognition of Persons by a Test of Statistical Independence*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 15, No 11, Nov. 1993.

[7] J. Daugman, *'How Iris Recognition Works'*, IEEE Transactions on Circuits and Systems For Video Technology, Vol. 14, n 1, January 2004.

[8] K. A. De Jong, *An Analysis of the Behaviour of a Class of Genetic Adaptive Systems.*, PhD thesis, University of Michigan, 1975

[9] A. de Santos Sierra, C. Sánchez Ávila, E. Marchiori, *Iris Recognition: Segmentation enhancement by using Morphological Operators*, Master Final Tesis, June, 2007. Amsterdam.

[10] A. de Santos Sierra, C. Sánchez Ávila, R. Sánchez Reíllo, *Sistema de identificación biométrica mediante patrón de iris utilizando operadores morfológicos y representación*, Congreso Iberoamericano de Seguridad Informática (CIBSI2007), November 2007, pags. 427-434, Mar del Plata, Argentina.

[11] A. E. Eiben, Z. Michalewicz, Eds. *Evolutionary Computation.*, IOS Press, 1998.

[12] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, Berlin, Germany: Springer, 2003.

[13] R. C. González, R. E. Woods, S. L. Eddins, *Digital Image Processing*, 2^{nd} Edn, Prentice All, 2004.

[14] J. H. Holland. *Adaptation in Natural and Artificial Systems*. MIT Press, Cambridge, MA, 1992, 1^{st} edition: 1975, The University of Michigan Press, Ann Arbor.

[15] J. H. Holland. Genetic algorithms and the optimal allocation of trials. *SIAM*, J. of Computing, 2 pp. 80-110, 1973.

[16] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolutionary Programs.*, Springer, Berlin, Heidelberg, New York, 3^{rd} edn., 1996.

[17] P. Rosenzweig, A. Kochems, and A. Schwartz, *Biometric Technologies: Security, Legal and Policy Implications*, published by The Heritage Foundation, N0. 12, June 21, 2004.

[18] R. Sánchez Reíllo, *El Iris ocular como parámetro para la Identificación Biométrica*, Ágora Sic Divulgación, vol. 2, Sep. 2000.

[19] H. P. Schwefel, *Evolution and Optimum Seeking*, Wiley, New York, 1995.

[20] CASIA Iris Image Database. http://www.sinobiometrics.com