# HUMAN IRIS CHARACTERIZATION THROUGH OPTIMUM-PATH FOREST AND DYNAMIC FEATURES

*R. M. da Costa*

Institute of Informatics
Federal University of Goiás
ronaldocosta@inf.ufg.br

*B. A. Gonzaga*

School of Engineering at São Carlos
University of São Paulo
small adilson@sel.eesc.usp.br

*J. P. Papa, A. N. Marana*

Department of Computing
Univ Estadual Paulista (UNESP)
{papa,nilceu}@fc.unesp.br

## ABSTRACT

Liveness detection methods have been extensively addressed to avoid frauds in biometric systems. In this paper, we introduce a recently method for the characterization of iris dynamics in the context of image classification, by assessing its performance with a novel pattern recognition technique called Optimum-Path Forest, which demonstrated to be similar to Self Organizing Maps, but much faster, and superior to Support Vector Machines and Artificial Neural Networks using Multilayer Perceptrons.

***Index Terms*—** Iris recognition, Biometrics, Optimum-Path Forest, Pattern Recognition

## 1. INTRODUCTION

The seminal work about human iris was initially proposed by Frank Burch in 1936. However, it was only with John Daugman that computerized applications were developed based on human identification assessing their iris properties [1]. Although several methods have been developed in the last decade for recognition based on iris features, most of the them are based on the algorithm created by Daugman [1]. The identification techniques are extremely precise, completing the process in a split second. However, all literature-known methods are applied to static images. In such a way, by observing the current methods, it is verified that they do not guarantee that the individual being evaluated is actually present or if the captured image is a photograph, prosthesis or some type of digital image, which may serve as entry data to the recognition system for frauds.

The art of distinguishing real people from photos or non real subjects receives the name of liveness detection, which has been extensively used to avoid frauds in biometric systems. Jee et al. [2], for instance, applied a methodology to detect real faces from non real ones. Kanematsu et al. [3] proposed a liveness detection method to detect fake iris from real ones using a variation in the brightness of an iris pattern induced by a pupillary reflex. Recently, Costa and Gonzaga [4] proposed an innovating method composed by an electronic device and a software to capture the dynamic properties of the iris, such as pupil contraction and dilation time, which is very robust to frauds regarding fake and computerized created iris. However, this work did not evaluate the dynamic features of the iris in the context of classification, only for precision and recall.

Support Vector Machines (SVM) and Artificial Neural Networks (ANN) have been also used for iris recognition. The former was applied by Roy and Bhattacharya [5], and the latter was addressed by [6, 7], among others. Despite the use of these artificial intelligence techniques have been increasing, some flaws need to be revisited. An ANN with multi-layer perceptrons (ANN-MLP), for example, can address linearly and non-linearly separable problems, but not non-separable situations with maximum effectiveness [8]. As an unstable classifier, collections of ANN-MLP can improve its performance up to some unknown limit of classifiers. Although Self Organizing Maps (SOM) [8] have been also extensively used in several applications, the selection of its parameters, i.e., the number of neurons in the squared lattice and the number of iterations of the algorithm, is a hard task and extremely dependent of the application.

Support Vector Machines (SVM) have been proposed to overcome the problem, by assuming linearly separable classes in a higher-dimensional feature space [9]. Its computational cost rapidly increases with the training set size and the number of support vectors. As a binary classifier, multiple SVM are required to solve a multi-class problem. Another important question is that the assumption of separability may also not be valid in any space of finite dimension.

Recently, a novel framework for graph-based classifiers that reduce the pattern recognition problem as an optimum path forest computation (OPF) in the feature space induced by a graph were presented [10]. These kind of classifiers do not interpret the classification task as a hyperplanes optimization problem, but as a combinatorial optimum-path computation from some key samples (prototypes) to the remaining nodes. Each prototype becomes a root from its optimum-path tree and each node is classified according to its strongly connected prototype, that defines a discrete optimal partition (influence region) of the feature space. The OPF-based classifiers have some advantages with respect to the aforementioned classifiers: (i) one of them is free of parameters, (ii) they do not assume any shape/separability of the feature space and (iii) run training phase faster, which allows the development of real time applications for fraud detection in electricity systems.

This work has as the main goal to characterize human iris by means of an OPF classifier, and has two main contributions: (i) to be the first into applying the OPF classifier for iris recognition and to be pioneer into using the dynamic iris' method proposed by Costa and Gonzaga [4] in the context of supervised classification. The remainder of this paper is organized as follows. Sections 2 and 3 present the OPF and the dynamic features of the iris method, respectively. Section 4 shows the experimental results, in which we compared OPF among two kind of Support Vector Machines, ANN-MLP and SOM networks. Section 5 states conclusions.

## 2. OPTIMUM-PATH FOREST

Let $Z_1$ and $Z_2$ be the training and test sets with $|Z_1|$ and $|Z_2|$ samples such as points or image elements. Let $\lambda(s)$ be the function that assigns the correct label $i$, $i = 1, 2, \ldots, c$, from class $i$ to any sample

$s \in Z_1 \cup Z_2$. $Z_1$ is a labeled set used to the design of the classifier and $Z_2$ is used to assess the performance of classifier and it is kept unseen during the project.

Let $S \subset Z_1$ be a set of prototypes of all classes (i.e., key samples that best represent the classes). Let $v$ be an algorithm which extracts $n$ attributes (color, shape or texture properties) from any sample $s \in Z_1 \cup Z_2$ and returns a vector $\vec{v}(s) \in \Re^n$. The distance $d(s,t)$ between two samples, $s$ and $t$, is the one between their feature vectors $\vec{v}(s)$ and $\vec{v}(t)$. One can use any valid metric (e.g., Euclidean) or a more elaborated distance algorithm. Our problem consists of using $S$, $(v,d)$ and $Z_1$ to project an optimal classifier which can predict the correct label $\lambda(s)$ of any sample $s \in Z_2$. The OPF classifier creates a discrete optimal partition of the feature space such that any sample $s \in Z_2$ can be classified according to this partition [10]. This partition is an optimum path forest (OPF) computed in $\Re^n$ by the image foresting transform (IFT) algorithm [11].

Let $(Z_1, A)$ be a complete graph whose the nodes are the samples in $Z_1$ and any pair of samples defines an arc in $A = Z_1 \times Z_1$. The arcs do not need to be stored and so the graph does not need to be explicitly represented. A path is a sequence of distinct samples $\pi = \langle s_1, s_2, \ldots, s_k \rangle$, where $(s_i, s_{i+1}) \in A$ for $1 \leq i \leq k-1$. A path is said *trivial* if $\pi = \langle s_1 \rangle$. We assign to each path $\pi$ a cost $f(\pi)$ given by a path-cost function $f$. A path $\pi$ is said optimum if $f(\pi) \leq f(\pi')$ for any other path $\pi'$, where $\pi$ and $\pi'$ end at a same sample $s_k$. We also denote by $\pi \cdot \langle s,t \rangle$ the concatenation of a path $\pi$ with terminus at $s$ and an arc $(s,t)$.

The OPF algorithm may be used with any *smooth* path-cost function which can group samples with similar properties [11]. We are interested in prototypes that fall in the region between classes, which are generally overlapped regions. So, we will address the path-cost function $f_{max}$, because of its theoretical properties for estimating prototypes that have this behavior (Section 2.1 gives the details about this procedure):

$$\begin{array}{rcl} f_{max}(\langle s \rangle) & = & \left\{ \begin{array}{ll} 0 & \text{if } s \in S, \\ +\infty & \text{otherwise} \end{array} \right. \\ f_{max}(\pi \cdot \langle s,t \rangle) & = & \max\{f_{max}(\pi), d(s,t)\}, \end{array} \qquad (1)$$

such that $f_{max}(\pi)$ computes the maximum distance between adjacent samples in $\pi$, when $\pi$ is not a trivial path.

The OPF algorithm assigns one optimum path $P^*(s)$ from $S$ to every sample $s \in Z_1$, forming an optimum path forest $P$ (a function with no cycles which assigns to each $s \in Z_1 \backslash S$ its predecessor $P(s)$ in $P^*(s)$ or a marker $nil$ when $s \in S$. Let $R(s) \in S$ be the root of $P^*(s)$ which can be reached from $P(s)$. The OPF algorithm computes for each $s \in Z_1$, the cost $C(s)$ of $P^*(s)$, the label $L(s) = \lambda(R(s))$, and the predecessor $P(s)$, as follows.

**Algorithm 1** – OPF ALGORITHM

| | |
|---|---|
| INPUT: | A $\lambda$-labeled training set $Z_1$, prototypes $S \subset Z_1$ and the pair $(v,d)$ for feature vector and distance computations. |
| OUTPUT: | Optimum-path forest $P$, cost map $C$ and label map $L$. |
| AUXILIARY: | Priority queue $Q$ and cost variable $cst$. |

```
1.  For each $s \in Z_1 \backslash S$, set $C(s) \leftarrow +\infty$.
2.  For each $s \in S$, do
3.      $C(s) \leftarrow 0$, $P(s) \leftarrow nil$, $L(s) \leftarrow \lambda(s)$, and insert $s$ in $Q$.
4.  While $Q$ is not empty, do
5.      Remove from $Q$ a sample $s$ such that $C(s)$ is minimum.
6.      For each $t \in Z_1$ such that $t \neq s$ and $C(t) > C(s)$, do
7.          Compute $cst \leftarrow \max\{C(s), d(s,t)\}$.
8.          If $cst < C(t)$, then
9.              If $C(t) \neq +\infty$, then remove $t$ from $Q$.
10.             $P(t) \leftarrow s$, $L(t) \leftarrow L(s)$, $C(t) \leftarrow cst$
11.             Insert $t$ in $Q$.
```

Lines $1-3$ initialize maps and insert prototypes in $Q$. The main loop computes an optimum path from $S$ to every sample $s$ in a non-decreasing order of cost (Lines $4-10$). At each iteration, a path of minimum cost $C(s)$ is obtained in $P$ when we remove its last node $s$ from $Q$ (Line 5). Ties are broken in $Q$ using first-in-first-out policy. That is, when two optimum paths reach an ambiguous sample $s$ with the same minimum cost, $s$ is assigned to the first path that reached it. Note that $C(t) > C(s)$ in Line 6 is false when $t$ has been removed from $Q$ and, therefore, $C(t) \neq +\infty$ in Line 9 is true only when $t \in Q$. Lines $8-11$ evaluate if the path that reaches an adjacent node $t$ through $s$ is cheaper than the current path with terminus $t$ and update the position of $t$ in $Q$, $C(t)$, $L(t)$ and $P(t)$ accordingly.

### 2.1. Training

We say that $S^*$ is an optimum set of prototypes when Algorithm 1 minimizes the classification errors for every $s \in Z_1$. $S^*$ can be found by exploiting the theoretical relation between minimum-spanning tree (MST) and optimum-path tree for $f_{max}$. The training essentially consists of finding $S^*$ and an OPF classifier rooted at $S^*$.

By computing an MST in the complete graph $(Z_1, A)$, we obtain a connected acyclic graph whose nodes are all samples of $Z_1$ and the arcs are undirected and weighted by the distances $d$ between adjacent samples. The spanning tree is optimum in the sense that the sum of its arc weights is minimum as compared to any other spanning tree in the complete graph. In the MST, every pair of samples is connected by a single path which is optimum according to $f_{max}$. That is, the minimum-spanning tree contains one optimum-path tree for any selected root node.

The optimum prototypes are the closest elements of the MST with different labels in $Z_1$ (i.e., elements that fall in the frontier of the classes). By removing the arcs between different classes, their adjacent samples become prototypes in $S^*$ and Algorithm 1 can compute an optimum-path forest with minimum classification errors in $Z_1$. Note that, a given class may be represented by multiple prototypes (i.e., optimum-path trees) and there must exist at least one prototype per class.

### 2.2. Classification

For any sample $t \in Z_2$, we consider all arcs connecting $t$ with samples $s \in Z_1$, as though $t$ were part of the training graph. Considering all possible paths from $S^*$ to $t$, we find the optimum path $P^*(t)$ from $S^*$ and label $t$ with the class $\lambda(R(t))$ of its most strongly connected prototype $R(t) \in S^*$. This path can be identified incrementally, by evaluating the optimum cost $C(t)$ as

$$C(t) = \min\{\max\{C(s), d(s,t)\}\}, \; \forall s \in Z_1. \qquad (2)$$

Let the node $s^* \in Z_1$ be the one that satisfies Equation 2 (i.e., the predecessor $P(t)$ in the optimum path $P^*(t)$). Given that $L(s^*) = \lambda(R(t))$, the classification simply assigns $L(s^*)$ as the class of $t$. An error occurs when $L(s^*) \neq \lambda(t)$.

## 3. DYNAMIC FEATURES FOR IRIS CHARACTERIZATION

The human eye is very sensitive to visible light. That is, the light within the violet-red range causes some type of reaction to the eye, from the cones and rods up to the sclera (the white outer part). For example: the pupil contracts and dilates under the effect of the visible light, and the iris and the sclera exceptionally reflect within this range. In order to capture an image of the human iris by using visible light, there is a problem: how to keep the natural reflexes on the globe of the eye, iris and sclera surfaces from affecting the quality of the digitalized image? Several techniques are employed by professional photographers in order to deviate the light beam, by appropriately positioning the camera. Still, to acquire iris images at a good resolution  thus allowing for the extraction of features aiming at the biometric recognition  the photograph techniques cannot be utilized because, in general, the camera must be placed frontally to the iris and at a short distance. These images do not provide enough quality for a dependable biometric recognition.

The near infrared (NIR) illumination generates good resolution and definition images. However, due to the fact that they are not ”visible” to the human eye, they do not allow for the necessary stimuli so that the pupil can perform the contraction and the dilation movements. The visible light offers the necessary stimulus. Nevertheless, the image quality is compromised, thus making the extraction of features difficult. In short: how to capture images with NIR illumination using visible light to contract and dilate the pupil, without causing reflexes on the iris, thus extracting the dynamic features?

Costa and Gonzaga [4] addressed this problem by taking into account the human optic system anatomy [1]. The eye captures through the cones and rods the light stimuli taken to the brain by the optic nerve, so that vision can be processed. In the transmission of the stimuli by the optic nerve, they pass through an area named optic chiasm. In this area, the mixture of the medium fibers of the optic nerve takes place. Fibers of the right optic nerve mix with the left ones and vice-versa. This causes the eyes to be connected, that is, the reflexes to stimuli applied to one of the eyes are presented in both. Such physiological function is denominated Consensual Reflex. This reflex is responsible for the synchronism of the movements for both eyes.

In such a way, Costa and Gonzaga [4] developed a device which performs different and independent tasks in each one of the eyes. The left eye receives visible light stimuli (white) in specific periods of time, controlled by the software developed, whereas the right eye image is digitalized in a video sequence, under NIR illumination. The acquisition device, connected to a computer, allows for the software to control the illumination on an eye, applied to in specific periods of time, whereas the image of the other eye is digitalized, thus forming a video sequence. The next section describes the method of Costa and Gonzaga [4] to extract and to select the dynamic features.

### 3.1. Feature extraction

In order to enable the evaluation of the behavior of each one of the features due to the illumination alteration conditions, five analysis periods are established. The first one is composed of the entire video (1,000 frames). The other periods are defined to record the precise period of illumination transition. This way, the periods comprise the following frames: (i) 1st period  average of the 1,000 frames, (ii) 2nd period  average between the frames 210 and 220, (iii) 3rd period  average between the frames 420 and 430, (iv) 4th period  average between the frames 630 and 640 and (v) 5th period  average between the frames 840 and 850.

Based on that, twelve different dynamic features are extracted: 1) Pupil circularity; 2) Pupil diameter; 3) Pupil contraction/dilation time; 4) Pupil contraction/dilation rate; 5) Average of the gray levels of the segmented iris; 6) Standard deviation of the gray levels of the segmented iris; 7) Variation coefficient of the gray levels of the segmented iris; 8) Correlation; 9) Angular Second Moment (ASM); 10) Entropy; 11) Contrast; 12) Inverse Difference Moment (IDM);

These twelve features above form the base to generate a feature vector, and for each one of them the average is calculated within each period. Costa and Gonzaga [4] argued that some performed tests with the features confirmed the feasibility of the dynamic features. However, a selection of the most discriminative and dynamic features was performed through the Weka (Waikato Environment for Knowledge Analysis) software [12], on order to reduce the dimensionality of the feature vector, which was originally composed by 248 features.

## 4. EXPERIMENTAL RESULTS

We performed two series of experiments: in the former (Section 4.1) we used 50% of the whole dataset for training and the remaining 50% for testing classifiers, and in the last one (Section 4.2) we executed the experiments with different training and test set size percentages to allow a comparison about the robustness of the classifiers with respect to variations on the sets size. For both experiments, we executed OPF, SVM-RBF (SVM with RBF as kernel function), SVM-LINEAR (SVM without kernel mapping), ANN-MLP (ANN-MLP trained by backpropagation algorithm) and Kohonen Self Organizing Maps (SOM) 10 times with randomly generated training and test sets, to compute the mean accuracy and its standard deviation, and the mean training and test execution times (in seconds).

For SVM-RBF, we used the latest version of the LibSVM package [13] with Radial Basis Function (RBF) kernel, parameter optimization and the one-versus-one strategy for the multi-class problem. With respect to SVM-LINEAR, we used the LibLINEAR package [14] with $C$ parameter optimized by cross-validation. For OPF we used the LibOPF [15], which is a library for the design of optimum-path forest-based classifiers, and for ANN-MLP we used the Fast Artificial Neural Network Library (FANN) [16]. The network configuration is $i$:$h_1$:$h_2$:$o$, where $i = 22$ (number of features), $h_1 = h_2 = 8$ and $o = 2$ (number of classes) are the number of neurons in the input, hidden and output layers, respectively. The ANN-MLP was trained with a backpropagation algorithm, and its architecture was empirically chosen. For SOM network we used a lattice with 100×100 neurons with 10 iterations for algorithm convergence.

### 4.1. Classifiers evaluation

We evaluate here the OPF, SVM-RBF, SVM-LINEAR, ANN-MLP and SOM for dynamic iris recognition using 50% for training and 50% for testing. Table 1 shows the mean accuracies and mean training and classification times (in seconds) after 10 runnings with randomly generated training and test sets.

**Table 1**. Mean accuracy and mean training and classification times OPF, SVM-RBF, SVM-LINEAR, ANN-MLP and SOM.

| Classifier | Accuracy | Training time | Classif. Time |
|---|---|---|---|
| **OPF** | **83.68±1.13** | **0.01** | **0.017** |
| SVM-RBF | 76.95±0.81 | 259.59 | 0.295 |
| SVM-LINEAR | 79.42±0.75 | 12.24 | 0.017 |
| ANN-MLP | 51.86±0.81 | 435.49 | 0.002 |
| SOM | 82.86±1.38 | 103.75 | 0.253 |

We can see that OPF and SOM obtained similar results if we take into account the standard deviation, and outperformed all remaining classifiers. However, the OPF classifier was much faster than SOM: 6900 times faster for training and 14.88 times faster for classification. This results confirmed that OPF is similar to the state of the art classification methodologies, but much faster [10].

### 4.2. Classifiers robustness

We evaluate the robustness of the classifiers with respect to variations on the training set size. We repeated the experiments shown in previous section with different training and test set sizes for OPF, SVM-RBF, SVM-LINEAR, ANN-MLP and SOM. Figure 1 displays the mean accuracies over the test set after 10 rounds of experiments for each training and test set size percentages configuration.

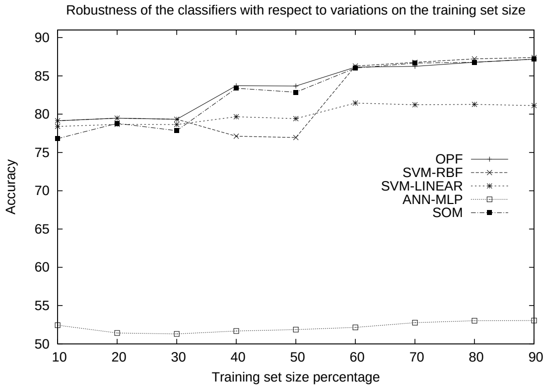

**Fig. 1**. Robustness of the classifiers with respect to different training and test size percentages.

We can see that OPF, SVM-RBF and SOM performed similar curves, except for SVM-RBF with 40%-50% of the training set size. We can see that ANN-MLP presented the worst results. One probably solution for that should be to use a collection of neural networks, although this is not guaranteed to solve the problem as a whole.

## 5. CONCLUSIONS

In this paper we introduced a recently developed dynamic iris characterization in the context of iris classification. Another main contribution concerns with the OPF first usage for automatic iris recognition. Experimental results comparing OPF among SVM-RBF, SVM-LINEAR, ANN-MLP and SOM have demonstrated the similarity between OPF and SOM in terms of effectiveness for this purpose, but the former was much faster both for training and classification phases.

In order to assess the robustness of the classifiers regarding variations on the training set size, we also performed an extra experiment with different percentages of the training set size. In this case, the OPF, SVM-RBF and SOM demonstrated to be similar, except for SVM-RBF with 40%-50% of training set size. Future works will be guided by using OPF in the context of static iris recognition.

## 6. REFERENCES

[1] John Daugman, "How iris recognition works," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 14, no. 1, pp. 21–30, 2004.

[2] Hyung-Keun Jee, Sung-Uk Jung, and Jang-Hee Yoo, "Liveness detection for embedded face recognition system," *World Academy of Science, Engineering and Technology*, vol. 16, pp. 29–32, 2006.

[3] M. Kanematsu, H. Takano, and K. Nakamura, "Highly reliable liveness detection method for iris recognition," in *Proceedings of the Annual Conference SICE*, 2007, pp. 361–364.

[4] R. M. da Costa and A. Gonzaga, "Extraction and selection of dynamic features of the human iris," in *Proceedings of the XXII Brazilian Symposium on Computer Graphics and Image Processing*. 2009, pp. 1–7, IEEE Press.

[5] Kaushik Roy and Prabir Bhattacharya, "Iris recognition with support vector machines," in *Advances in Biometrics*. 2005, vol. 3832, Lecture Notes in Computer Science, Springer Berlin / Heidelberg.

[6] Ahmad M. Sarhan, "Iris recognition using discrete cosine transform and artificial neural networks," *Journal of Computer Science*, vol. 5, no. 5, pp. 369–373, 2009.

[7] Muhammad Sarfraz, Mohamed Deriche, Muhammad Moinuddin, and Syed Saad Azhar Ali, "Intelligent iris recognition using neural networks," *Computer-Aided Intelligent Recognition Techniques and Applications*, pp. 145–167, 2005.

[8] S. Haykin, *Neural networks: a comprehensive foundation*, Prentice Hall, 1994.

[9] C. Cortes and V. Vapnik, "Support vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.

[10] J.P. Papa, A.X. Falcão, and C.T.N. Suzuki, "Supervised pattern classification based on optimum-path forest," *International Journal of Imaging Systems and Technology*, vol. 19, no. 2, pp. 120–131, 2009.

[11] A.X. Falcão, J. Stolfi, and R.A. Lotufo, "The image foresting transform theory, algorithms, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 19–29, 2004.

[12] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten, "The weka data mining software: An update," *SIGKDD Explorations*, vol. 11, no. 1, 2009.

[13] C. C. Chang and C. J. Lin, *LIBSVM: A Library for Support Vector Machines*, 2001, Software available at url http://www.csie.ntu.edu.tw/˜ cjlin/libsvm.

[14] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[15] J.P. Papa, C.T.N. Suzuki, and Alexandre Xavier Falcão, *LibOPF: A library for the design of optimum-path forest classifiers*, 2009, Software version 1.0 available at `http://www.ic.unicamp.br/~afalcao/LibOPF`.

[16] S. Nissen, *Implementation of a Fast Artificial Neural Network Library (FANN)*, 2003, Department of Computer Science University of Copenhagen (DIKU). Software available at *http://leenissen.dk/fann/*.