Iris Biometrics: Indexing and Retrieving Heavily Degraded Data

Hugo Proença, *Member, IEEE*

***Abstract*—Most of the methods to index iris biometric signatures were designed for decision environments with a clear separation between genuine and impostor matching scores. However, in case of less controlled data acquisition, images will be degraded and the decision environments poorly separated. This paper proposes an indexing / retrieval method for degraded images and operates at the *code* level, making it compatible with different feature encoding strategies. Gallery codes are decomposed at multiple scales, and according to their most reliable components at each scale, the position in an n-ary tree determined. In retrieval, the probe is decomposed similarly, and the distances to multi-scale centroids are used to penalize paths in the tree. At the end, only a subset of the branches is traversed up to the last level. When compared to related strategies, the proposed method outperforms them on degraded data, particularly in the performance range most important for biometrics (hit rates above 0.95). Finally, according to the computational cost of the retrieval phase, the number of enrolled identities above which indexing is computationally cheaper than an exhaustive search is determined.**

***Index Terms*—Iris Recognition, Indexing / Retrieval, Wavelet Decomposition / Reconstruction.**

I. Introduction

Biometrics is used in various scenarios with satisfactory results (e.g., refugee control, security assessments and forensics). Among various traits, the iris is one of the most popular, due to several important properties: 1) it is an internal organ, naturally protected and visible from the exterior; 2) its texture has a randotypic chaotic appearance that is apparently stable over human lifetime; 3) it provides high accuracy and extreme robustness against false matches, even in national-scale scenarios; and 4) it enables real-time data processing, mainly due to the binary nature of iris codes.

The nationwide deployment of iris recognition systems is considered a success. In the last information update about the UAE system [2], over 2 million identities were included on its watch-list, and more than 350,000 deportees were prevented from entering the Emirates. The Unique Identification Authority of India [17] is deploying the system at the largest scale, with more than 300 million persons enrolled and adding about one million new identities per day, performing 6e^{14} daily cross-comparisons to search for duplicate identities.

According to the most acknowledged iris recognition method [1], matching *IrisCodes* primarily involves the accumulation of bitwise XOR operations. However, despite the extreme computational effectiveness of this matching scheme, the time required for exhaustive searches grows linearly with the number of enrolled identities. Also, the time required for de-duplication searches grows quadratically with respect to the size of the database, which is specially concerning in nationwide scales.

As noted by Hao *et* al. [6], indexing is a specific case of the general *nearest neighbor search* problem, and motivated several proposals in the last years (section II). However, most of these methods were designed for decision environments of good quality, with a clear separation between the genuine and impostor matching scores.

In this paper, we are mainly interested in decision environments with a significant overlap between the genuine and impostors matching scores, corresponding to systems that operate in less controlled data acquisition protocols. We propose an indexing / retrieval method that runs at the code level, i.e., after the feature encoding process. Codes are decomposed in a coarse-to-fine scheme, and their position in an n-ary tree determined. In retrieval, the probe is decomposed in a similar way, and the distances to multi-scale centroids are obtained, penalizing most of the paths in the tree and traversing only a subset of nodes down to the leaves. When compared to related works, the main contributions of the proposed method are three-fold: 1) it is compatible with different signature encoding methods; 2) it outperforms the state-of-the-art approaches in poor quality data, particularly in the performance range that is meaningful for biometrics (hit rates above 0.95); and 3) it has a reduced computational cost: compared to exhaustive searches, indexing becomes advantageous when more than a few thousands identities are enrolled.

The suitability of the proposed method to handle degraded data has its roots in the concepts of *coarse-to-fine analysis* (in indexing) and *non-excluding branches* (in retrieval): 1) in indexing, *IrisCodes* are grouped in branches of the tree according to their multi-scale features, being *tree-level* and *analyzed-scale* in direct relationship. This means that at the root (maximum) level, *IrisCodes* are grouped according to their lowest frequency components. 2) in retrieval, a parallel searching scheme was devised: starting with a residual value, the tree is traversed along different branches until the sum of residual penalizations for each branch guarantees that the identity of interest will not be found there.

The remainder of this paper is organized as follows: Section II summarizes the most relevant iris indexing strategies published. Section III provides a description of the proposed method. Section IV presents and discusses the results with re-

Author is with the Department of Computer Science, IT - Instituto de Telecomunicações, University of Beira Interior, 6200-Covilhã, Portugal. hugomcp@di.ubi.pt

Manuscript received ? ?, ?; revised ? ?, ?.

spect to the state-of-the-art techniques. Finally, the conclusions are given in Section V.

II. Related Work

Indexing / retrieving a biometric identity in a database is a particularly sensitive task, as failures compromise the subsequent processing and lead to matching errors. Upon a *query*, the indexing techniques attempt to maximize the number of times that the *identity-of-interest* is included among a group of returned identities (*hit* rate), while maintaining that group as small as possible (*penetration* rate).

Table I summarizes the iris indexing methods reported in the literature, which can be coarsely classified using two criteria: 1) the light spectrum used in data acquisition (either at near-infrared or visible wavelengths); and 2) the input, which is either the iris texture or the biometric signature (*IrisCode*). Also, we summarize our viewpoint with respect to each method in the column ”Pro. / Cons.”, emphasizing the most relevant advantages (underlined) and drawbacks (regular font) of each one.

Yu *et* al. [19] represent the iris data in the polar domain, divided radially into sixteen regions, and obtain the fractal dimension for each one. Using first-order statistics, a set of semantic rules indexes the data into one of four classes. In retrieval, each probe is matched exclusively against gallery data in the same class. Fu *et* al. [4] use color information and suggest that artificial color filters provide an orthogonal discriminator of the spatial iris patterns. Each filter is represented by a discriminator that operates at the pixel level. Gadde *et* al. [5] analyze the distribution of intensities and select patterns with low coefficients of variation (CV) as indexing pivots. For each image in the polar domain, a radial division of n-bands is performed and the highest densities of CV patterns considered. Hao *et* al. [6] exclusively analyze the *IrisCodes* and how their most reliable bits spread, based on the notion of multi-collisions. In retrieval, a minimum of k collisions between the probe and gallery codes is required to consider a potential match. Jayaraman and Prakash [9] fuse texture and color information: they estimate the color of the iris in the YCbCr space and determine an index to reduce the search space. Texture is encoded by the SURF technique. Mehrotra *et* al. [11] use SIFT descriptors and their spatial distribution. To overcome the effect of non-uniform illumination and partial occlusions, keypoints are extracted from angularly constrained regions of the iris. In retrieval, the geometric hashed location of keypoints determines the bin of a hash table, casting a vote per entry. The most voted identities are considered the possible candidates. Mehrotra *et* al. [12] divide the polar iris data into bands using a multi-resolution DCT transformation. Energy-based histograms are extracted from these bands, divided into fixed-size bins, and irises with similar energy values are grouped. A B-tree is built, in which leaves contain elements with the same key. For a query, the corresponding key is generated, and the tree traversed until a leaf node is reached. The templates stored at that leaf constitute the set of potential identities. Mukherjee and Ross [13] address the problem from two different perspectives: by analyzing the iris texture and the *IrisCode*. The best results in the latter case are attained when each code is divided into fixed-size blocks. First-order statistics for each block are used as primary indexing value. A k-means strategy divides the feature space. Qiu *et* al. [16] create a small dictionary of visual words (*textons*), to represent the visual primitives of iris images. Then, texton histograms are the global features, and the k-means algorithm groups elements into five categories. Vatsa *et* al. [18] represent pixels of the unwrapped iris data in an 8-D binary feature space. The most significant bits are used to build four maps from which the Euler numbers are extracted. Retrieval is done according to the nearest neighbor technique. Zhao [20] obtains the average RGB values inside the iris, weighted by the luminance component, and project these values into independent 1-D spaces. Probes are matched against gallery elements in the union of identities inside bins of these spaces. A similar approach is due to Puhan and Sudha [15]: they obtain the color index (in the YCbCr color space) and use a semantic decision tree to index the database.

Table I enables to perceive how heterogeneous are the techniques used in indexing / retrieval proposals. It is interesting to note that most techniques use data acquired at visible wavelengths, which is justified by the analysis of color-based features. However, this kind of techniques lack in terms of generalization, as they cannot be used in near-infrared data. Also, some of these methods use keypoints-based feature descriptors (e.g., SIFT, SURF), which might be problematic due to two reasons: the significant computational cost of these techniques and their sensitivity to slight changes in focus, that are frequent in iris data.

III. Proposed Method

A. Indexing

1) Codes Decomposition / Reconstruction: Let $\boldsymbol{s}_i$ denote an *IrisCode* from the ith subject. As illustrated in Figure 1, the rationale is to obtain coarse-to-fine representations of $\boldsymbol{s}$ as a function of the level l in the tree ($\boldsymbol{s}^{(l)}$). These representations are grouped according to their similarity in the L_2 metric space, and stored in each node. A node is considered a leaf when its centroid $\boldsymbol{c}$ is at a sufficiently small distance from all elements, i.e., $||\boldsymbol{s}_i^{(l)} - \boldsymbol{c}||_2 \leq \nu$, $\forall i$.

Let $\phi(x) = \sum_{k\in\mathbb{Z}} h(k)\sqrt{2}\phi(2x-k)$ and $\psi(x) = \sum_{k\in\mathbb{Z}} g(k)\sqrt{2}\phi(2x-k)$, where $\boldsymbol{h}(.)$ and $\boldsymbol{g}(.)$ are low-pass and high-pass filters. According to Mallat's multiresolution analysis [10], the operator representation of these filters is:

$$\begin{aligned} \boldsymbol{H}_a^{(k)} &= \sum_n h(n-2k)a_n \\ \boldsymbol{G}_a^{(k)} &= \sum_n g(n-2k)a_n, \end{aligned} \tag{1}$$

where $\boldsymbol{H}_a^{(k)}$ and $\boldsymbol{G}_a^{(k)}$ are one-step wavelet decompositions. Let len$(\boldsymbol{s}^{(n)}) = N = 2^n$ be the length (in our experiments, $n = 11$) of the signal $\boldsymbol{s}$ represented at scale n by a linear combination of ϕ filters:

TABLE I
OVERVIEW OF THE MOST RELEVANT RECENTLY PUBLISHED IRIS INDEXING METHODS. *NIR* STANDS FOR NEAR-INFRARED AND *VW* FOR VISIBLE WAVELENGTH DATA.

Method	**Type**	**Data (Spectrum)**	**Preprocessing**	**Summary**	**Pro. / Cons.**
Fu *et* al. [4]	Color	Own (VW, 9 images)	Segmentation	Artificial color filters, pre-tuned to a range of colors. C-means to define classes. Pixel-by-pixel Euclidean distance to clusters used in indexing	Computational cost, Independent of the iris encoding technique; exclusive for color data, requires high-quality data, experiments on a reduced dataset
Gadde *et* al. [5]	Texture, *IrisCode*	CASIA-V3 (NIR)	Segmentation, normalization	Estimation of intensity distribution, binarization, counting binary patterns with less coefficient of variation, division into radial bands, density estimation	compatible with different iris encoding techniques, robustness to changes in illumination; robustness to parameterization (number of bands)
Hao *et* al. [6]	*IrisCode*	632 500 UAE *IrisCodes*	Segmentation, normalization, feature extraction	Selection of most reliable bytes, bits decorrelation (interleaving and rotations), partition of identities into beacons, detection of multiple collisions	computational cost, robustness to parameterization; requires binary signatures
Jayaraman and Prakash [9]	Color, Texture	UBIRIS.v1, UPOL (VW)	Segmentation	Color analysis in YCbCr space. SURF keypoint description, Kd-tree indexing	Independent of the iris encoding technique; exclusive for color data, computational cost
Mehrotra *et* al. [11]	Texture	CASIA.1, ICE, WVU (NIR)	Segmentation	Keypoints localization, geometric analysis, hash table construction	Independent of the iris encoding technique, robustness to changes in rotation; computational cost; robustness to changes in focus
Mehrotra *et* al. [12]	Texture	CASIA, Bath, IITK (NIR)	Segmentation, normalization	Multi-resolution decomposition (DCT). Energy of subbands extracted in Morton order, B-tree indexing	Independent of the iris encoding technique ; computational cost; uneven analysis of iris data
Mukherjee and Ross [13]	Texture, *IrisCode*	CASIA-V3 (NIR)	Segmentation, normalization	1) sub-blocks division, top-n similarity between blocks, tree partition; 2) sublocks partition, k-means clustering	Independent of the iris encoding technique, computational cost; sensitive to parameter values (number clusters)
Puhan and Sudha [15]	Color	UBIRIS.v1, UPOL (VW)	Segmentation	Conversion to YCbCr, semantic decision tree	Independent of the iris encoding technique; exclusive for color data, sensitive to parameter values (number indexes)
Qiu *et* al. [16]	Texture	CASIA.1, ICE, WVU (NIR)	Segmentation, normalization	Extraction of texton histograms, Chi-square dissimilarity, K-means clustering	Independent of the iris encoding technique, robust to global changes in illumination; computational cost, sensitive to the number of categories chosen
Vatsa *et* al. [18]	Texture	CASIA.1, ICE, WVU (NIR), UBIRIS.v1 (VW)	Segmentation, normalization	8-bit planes of the masked polar image, extraction of topological information (Euler numbers), nearest neighbor classification	Independent of the iris encoding technique, computational cost; requires accurate segmentation, sensitive to iris rotation
Yu *et* al. [19]	Texture	CASIA.1, ICE, WVU (NIR)	Segmentation, normalization	Definition of radial ROIs, extraction of local fractal dimensions, semantic decision tree	Independent of the iris encoding technique; computational cost, sensitiveness to parameters (rules)
Zhao [20]	Color	UBIRIS.v2 (VW)	Segmentation, noise detection	Estimation of luminance, color compensation, average color, projection and quantization into three 1D feature spaces, union of identities from corresponding bins	Independent of the iris encoding technique; exclusive for color data, sensitiveness to parameters (interval length)

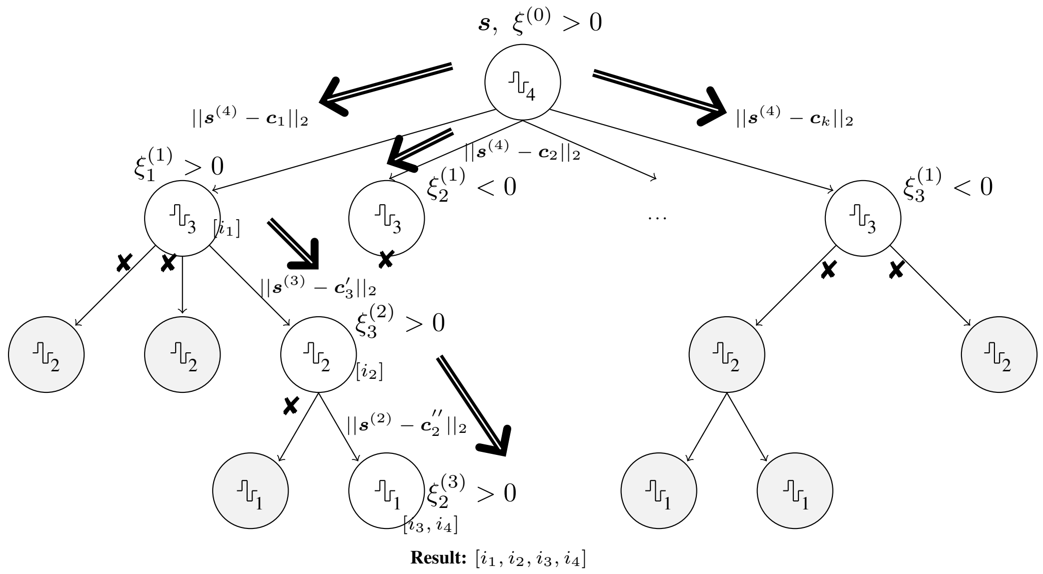

$\boldsymbol{s},\ \xi^{(0)} > 0$
$||\boldsymbol{s}^{(4)} - \boldsymbol{c}_1||_2$
$||\boldsymbol{s}^{(4)} - \boldsymbol{c}_2||_2$
$||\boldsymbol{s}^{(4)} - \boldsymbol{c}_k||_2$
$\xi_1^{(1)} > 0$
$\xi_2^{(1)} < 0$
$\xi_3^{(1)} < 0$
$\ldots$
$[i_1]$
$||\boldsymbol{s}^{(3)} - \boldsymbol{c}_3'||_2$
$\xi_3^{(2)} > 0$
$[i_2]$
$||\boldsymbol{s}^{(2)} - \boldsymbol{c}_2''||_2$
$\xi_2^{(3)} > 0$
$[i_3, i_4]$
Result: $[i_1, i_2, i_3, i_4]$

Fig. 1. Cohesive perspective of the indexing structure and of the retrieval algorithm. For a query $\boldsymbol{s}$ with *residual* $\xi^{(0)}$, the distance between the decomposition of $\boldsymbol{s}$ at top level ($\boldsymbol{s}^{(4)}$) to the centroids $\boldsymbol{c}_i$ is used to generate the new generation of residuals ($\xi_{.}^{(1)}$). For any branch with negative values, the search is stopped, meaning that subsequent levels in the tree are not traversed (illustrated by gray nodes). When traversing the tree, every identity found at any node while $\xi_{.}^{(\cdot)} > 0$ is included in the retrieved set.

$$\boldsymbol{s}^{(n)} = \sum_n a_k^{(n)} \phi_{nk}. \tag{2}$$

At each iteration, a coarser approximation $\boldsymbol{s}^{(j-1)} = \boldsymbol{H}\ \boldsymbol{s}^{(j)}, j \in \{1,\ldots,n\}$, is obtained: $\boldsymbol{d}^{(j-1)} = \boldsymbol{G}\ \boldsymbol{s}^{(j)}$ are the residuals of the transformation $\boldsymbol{s}^{(j)} \rightarrow \boldsymbol{s}^{(j-1)}$. The discrete wavelet transformation of $\boldsymbol{s}^{(n)}$ is:

$$\boldsymbol{s}^{(n)} \equiv [\boldsymbol{d}^{(n-1)}, \boldsymbol{d}^{(n-2)}, \ldots, \boldsymbol{d}^{(0)}, \boldsymbol{s}^{(0)}], \tag{3}$$

where $\left(\sum_{i=0}^{n-1} \text{len}(\boldsymbol{d}^{(i)})\right) + \text{len}(\boldsymbol{s}^{(0)}) = \text{len}(\boldsymbol{s}^{(n)}) = 2^n$.

In reconstruction, $\boldsymbol{s}^{(n)}$ can be approximated at different levels using $\boldsymbol{H}^*$ and $\boldsymbol{G}^*$ filters:

$$\begin{aligned} (\boldsymbol{H}_a^*)^{(n)} &= \sum_k h(n-2k)a_k \\ (\boldsymbol{G}_a^*)^{(n)} &= \sum_k g(n-2k)a_k, \end{aligned} \tag{4}$$

where $\boldsymbol{s}^{(n)} = \sum_{j=0}^{n-1} (\boldsymbol{H}_a^*)^{(j)} (\boldsymbol{G}_a^*)^{(j)} \boldsymbol{d}^{(j)} + (\boldsymbol{H}^*)^{(n)} (\boldsymbol{G}_a^*)^{(n)} \boldsymbol{s}^{(0)}$. Considering that *IrisCodes* are binary, the Haar wavelet maximally correlates them and its filter coefficients are $\boldsymbol{h} = [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]$, $\boldsymbol{g} = [\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}]$, with similar reconstruction coefficients $\boldsymbol{h}^* = \boldsymbol{h}$ and $\boldsymbol{g}^* = -\boldsymbol{g}$. Under this strategy, $\boldsymbol{H}$ acts as a smoothing filter and $\boldsymbol{G}$ as a detail filter.

When reconstructing a signal at a given level, the detail coefficients of small magnitude can be disregarded, as they intuitively do not have a major role in the signal. This is possible because wavelets provide an unconditional basis, i.e., one can determine whether an element is important by analyzing the magnitudes of the coefficients used in the linear combination of the basis vectors.

The threshold (λ) for the minimal magnitude of the coefficients considered was found according to the idea of *universal threshold*, due to Donoho and Johnstone [3]. Here, detail coefficients with a magnitude smaller than the expected maximum for an independent and identically distributed (Normal dist.) noise sequence were ignored:

$$\lambda = \sqrt{2\log(n)}\hat{\sigma}, \tag{5}$$

where 2^n is the length of the original signal and σ is given by:

$$\sigma = \sqrt{\frac{1}{N/2-1} \sum_{i=1}^{N/2} (d_i^{(l)} - \bar{d})^2}, \tag{6}$$

where $d_i^{(l)}$ denotes the ith wavelet coefficient at level l and $\bar{d}$ is the mean of coefficients. Figure 2 illustrates representations at different levels l, $(l \in \{0, 1, \ldots, 10\})$ of an *IrisCode* $\boldsymbol{s}$. The coarsest representation $\boldsymbol{s}^{(10)}$ retains the lowest frequency components of the signature (intensities are stretched for visualization purposes) and is used in the root of the tree. The finest representation $\boldsymbol{s}^{(0)}$ is used in the leaves.

As Figure 3 turns evident, $\boldsymbol{s}^{(l)}$ are increasingly smoothed versions of $\boldsymbol{s}$. The leftmost plot shows the average residuals between $\boldsymbol{s}$ and its reconstructions at level l (horizontal axis), being evident that residuals increase directly with respect to the decomposition level. The center and rightmost plots give histograms of the residuals for the coarsest (center) and finest scales (right), enabling to perceive that the reconstruction at the coarsest scale is essentially a mean of the original signal.

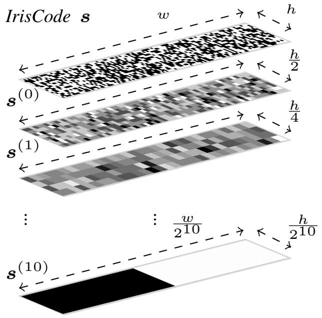


Fig. 2. Representation of an *IrisCode* (upper image) at different levels, retaining coarse (bottom image) to fine information from the input code. The $\boldsymbol{s}^{(10)}$ representation is used in the root of the tree and the remaining representations at the deeper levels. Intensities and sizes are stretched for visualization purposes.

2) Determining the Number of Branches per Node: Having a set of signals $\boldsymbol{s}^{(l)}$ in a node, a clustering algorithm finds their centroid $\boldsymbol{c}$ and creates a partition, according to the distances to that centroid. Also, if $||\boldsymbol{s}_i^{(l)} - \boldsymbol{c}||_2 \leq \nu, \forall i$, the process stops at level l for that branch, and the node is considered a leaf.

The number of clusters determines the number of branches in each node of the tree. In order to obtain the *optimal* value, a comparison between the proportion of variance in the data with respect to the number of clusters was carried out. Intuitively, for a too small number of clusters, new partitions reduce the variance significantly, but if the number of clusters is too large, adding a new one almost doesn't change variance. Hence, the number of clusters was considered optimal when this marginal gain decreases more significantly. Let k be the number of clusters. The proportion of the variance explained is characterized by a F-test:

$$F(k) = \frac{(m-k)\sum_{j=1}^{k} m_j ||\overline{\boldsymbol{s}_{(j)}} - \overline{\boldsymbol{s}}||_2}{(k-1)\sum_{j=1}^{k}\sum_{i=1}^{k} ||\boldsymbol{s}_{i(j)} - \overline{\boldsymbol{s}_{(j)}}||_2}, \tag{7}$$

where $\boldsymbol{s}_{i(j)}$ is the ith element in the jth cluster, $\overline{\boldsymbol{s}_{(j)}}$ is the sample mean in that cluster, m_j is the number of codes in a cluster ($m = \sum m_j$) and $\overline{\boldsymbol{s}}$ the overall mean. Considering $\big(k_t, F(k_t)\big)$ points on a curve, the value with minimal curvature corresponds to the number of clusters at which the marginal gain drops more. Parameterizing the curve $\big(x(t), y(t)\big) = \big(k_t, F(k_t)\big), t = \{1, 2, \ldots\}$, using quadratic polynomials:

$$\begin{cases} x(t) = a_3 t^2 + a_2 t + a_1 \\ y(t) = b_3 t^2 + b_2 t + b_1. \end{cases} \tag{8}$$

Using the previous $(t-1)$ and next $(t+1)$ points with respect to each position t, the least squares strategy is used to obtain the $a_.$ and $b_.$ coefficients:

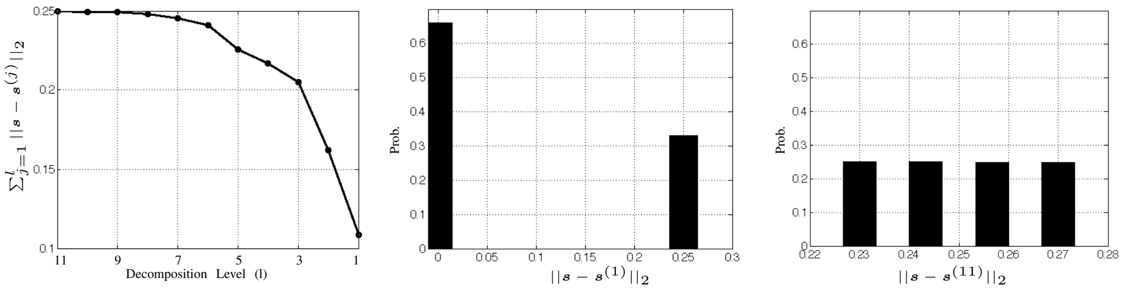


Fig. 3. Average sum of residuals between an *IrisCode* $\boldsymbol{s}$ and its representations at different levels ($\boldsymbol{s}^{(l)}$ (leftmost image). The images at the center and far right give the histograms of the residuals for decompositions/reconstructions at the finest (center) and coarsest (right) levels.

$$\Upsilon_a = \sum_{t_0=t-1}^{t+1} \Big(y_{t_0} - \big(a_1 + a_2 x_{t_0} + a_3 x_{t_0}^2\big)\Big)^2. \quad (9)$$

Setting $\frac{\partial \Upsilon}{\partial a_j} = 0$ yields

$$\begin{bmatrix} 3 & \sum x_{t_0} & \sum x_{t_0}^2 \\ \sum x_{t_0} & \sum x_{t_0}^2 & \sum x_{t_0}^3 \\ \sum x_{t_0}^2 & \sum x_{t_0}^3 & \sum x_{t_0}^4 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} \sum y_{t_0} \\ \sum x_{t_0} y_{t_0} \\ \sum x_{t_0}^2 y_{t_0} \end{bmatrix} \quad (10)$$

By solving the system of linear equations for $a_.$, the coefficients of the polynomials are found (and is analogous for $b_.$ values). The curvature κ at each point k_t corresponds to:

$$\kappa(k_t) = \frac{x(t)' y(t)'' - y(t)' x(t)''}{\sqrt{(x(t)'^2 + y(t)'^2)^3}}, \quad (11)$$

where primes denote derivatives with respect to t. In our case, $x'(t) = 2ta_3 + a_2$, $x''(t) = 2a_3$, $y'(t) = 2tb_3 + b_2$ and $y''(t) = 2b_3$. Hence, (11) can be written as:

$$\kappa(k_t) = \frac{(2ta_3 + a_2)2b_3 - 2a_3(2b_3 + b_2)}{\big((2ta_3 + a_2)^2 + (2tb_3 + b_2)^2\big)^{\frac{3}{2}}}. \quad (12)$$

Because we are primarily interested in the curvature at each point, t can be set to 0, obtaining:

$$\kappa(k_t) = \frac{2(a_2 b_3 - a_3 b_2)}{(a_2^2 + b_2^2)^{\frac{3}{2}}}. \quad (13)$$

Finally, the position with minimal curvature is deemed to be the optimal number of clusters for that node:

$$\hat{k} = \arg\min_t \kappa(k_t). \quad (14)$$

Figure 4 shows an example of the method that finds the number of clusters. Here, the $F(k_t)$ values were obtained for $t \in \{2, \ldots, 11\}$ (continuous lines). The dashed line corresponds to the $\kappa(k_t)$ values. The minimum curvature of the interpolating polynomials was observed at $\hat{k} = 8$. In this case, $F(k_t)$ bends clockwise when the proportion of explained variance no longer increases. According to (14), this is the point of minimal curvature and is used as the number of clusters for a node.

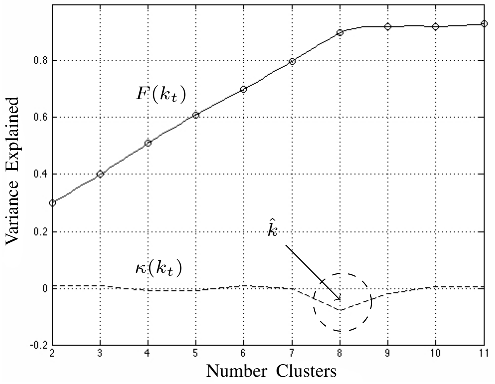


Fig. 4. Illustration of the strategy to determine the number of clusters at each node. For ($k_t \in \{2, \ldots, 11\}$), the amount of variance explained $F(k_t)$, is denoted by circular data points. Quadratic polynomials were fitted to interpolate this data (continuous lines), from where the curvature at each point was found (dashed line). The number of clusters $\hat{k} = 8$ corresponds to the point where the gain in the explained variance no longer increases, i.e., where the curvature value $\kappa(k_t)$ attains a minimum.

B. Retrieval

As below described, one of the most relevant properties of the proposed method is its *non-exclusiveness* in terms of the paths in the tree traversed for a query. This *parallel* searching scheme is particularly important for our purposes, as it contributes for the robustness against degraded data.

Retrieval requires a query signature $\boldsymbol{s}$ and a residual value $\xi > 0$. The idea is to iteratively decrease the residuals for each branch (ξ_j is the residual for the jth branch of a node) and stop when $\xi_j < 0$. At each node, the ℓ_2 distance between the reconstructed signal $\boldsymbol{s}^{(l)}$ and a cluster centroid $\boldsymbol{c}_i$ is subtracted from ξ_j, considering the maximum distance between $\boldsymbol{c}_i$ and all the identities in that branch. Formally, let $q(\boldsymbol{s}, \xi^{(0)})$ be the query parameters. Let $\boldsymbol{s}^{(l)}$ be the reconstruction of $\boldsymbol{s}$ at level l. The next generation of residual values $\xi^{(l+1)}$ is given by:

$$\xi^{(l+1)}_{.} = \xi^{(l)} - \max\Big(0, ||\boldsymbol{s}^{(l)} - \boldsymbol{c}_i^{(l)}||_2 - \max\big(||\boldsymbol{s}_b^{(l)} - \boldsymbol{c}_i^{(l)}||_2, \forall b \in \{1,\dots,t_i\}\big)\Big), \quad (15)$$

being $\boldsymbol{c}_i^{(l)}$ the i^{th} cluster at level l and $\boldsymbol{s}_b^{(l)}$ the remaining signatures (total t_i) in that branch of the tree. The set of identities retrieved is given by:

$$q(\boldsymbol{s}, \xi^{(l)}) = \begin{cases} [\{i_.\}, q(\boldsymbol{s}, \xi_j^{(l+1)})], \forall j, & \text{, if } \xi^{(l)} > 0 \wedge l > 1 \\ \{i_.\} & \text{, if } \xi^{(l)} > 0 \wedge l = 1 \\ \emptyset & \text{, if } \xi^{(l)} \leq 0 \end{cases} \quad (16)$$

where $[,]$ denotes vector concatenation, $\xi_j^{(l)}$ denotes the residual value for the jth branch at level l and $\{i_.\}$ is the set of identities in a node.

Due to the intrinsic properties of wavelet decomposition, the distance values at the higher scales should be weighted by $w()$, as they represent more signal components:

$$w(l) = \frac{1 + \text{erf}\Big(\alpha(l-n)\Big)}{2}, \quad (17)$$

being α a parameter that controls the shape of the sigmoid. Figure 5 shows one example of the histograms of the cuts in residuals ($\xi^{(l)} - \xi^{(l+1)}$, horizontal axis) with respect to the level in the tree. The dashed vertical lines indicate the cuts in the path that contained the identity of interest. Note that, with exception of the leaf level ($l = 1$), no cuts in the residuals were performed for the *interesting* path. This is in opposition to the remaining paths, where cuts occurred at all levels.

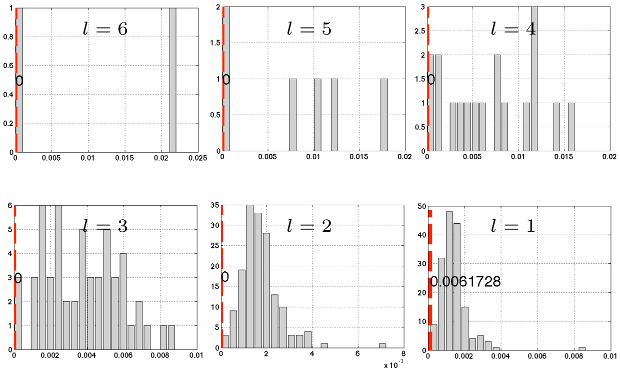


Fig. 5. Histograms of the cuts in residuals $\xi^{(l)} - \xi^{(l+1)}$ per level during retrieval. The vertical dashed lines give the cumulative distribution values of the cuts in the path that contains the identity of interest. Gray bars express frequencies of the cuts in the remaining paths of the tree.

C. Time complexity

Here we are mainly interested in the time complexity of the retrieval algorithm, and how the turnaround time depends on the number of identities enrolled. Let t_s, t_c and t_m denote the average elapsed time in the iris segmentation, feature coding and matching stages. Without indexing, the average turnaround time for an exhaustive search t_e is given by

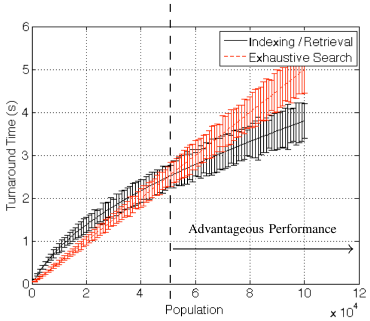
$$t_e = t_s + t_c + N\ 0.5\ t_m, \quad (18)$$

where N is the number of identities enrolled by the system. When indexing at the *IrisCodes* phase, the average turnaround time t_i corresponds to

$$t_i = t_s + t_c + N\ t_r + \Big(h\ p + (1-h)\Big)\ 0.5\ N\ t_m, \quad (19)$$

being t_r the average turnaround time for retrieval and h and p the hit and penetration rates.

The left plot in Figure 6 compares the values for the t_i and t_e turnaround times with respect to the number of identities enrolled, when using the proposed method. t_s and t_c were disregarded because they do not affect the comparison. The values were obtained by repeatedly assessing the turnaround times of the proposed method and of exhaustive searches. The horizontal bars near each point give the range of values observed, enabling to conclude that the proposed method starts to be advantageous when more than 54,000 identities are enrolled (vertical dashed line). Note that this value depends of the hit / penetration rates considered, which are function of the data quality. Even though, it is an approximation of the minimum number of identities that make indexing advantageous in terms of turnaround time. Also, the Table given at the right part of Figure 6 compares the expected values for t_i and t_e (in seconds) for typical scenarios: local, national, continental and global scales. These values stress the importance of indexing for operating at large scales.



Scenario	**Tot. Ids.**	t_e (s)	t_i (s)
Local	$10e^5$	4.95	3.74
National	$10e^7$	495	58.9
Continental	$10e^9$	$49e^3$	927
Global	$7 \times 10e^9$	$346e^3$	$297e^1$

Fig. 6. At left: comparison between the turnaround times of an exhaustive search (red line) and when using the proposed indexing / retrieval strategy (black line), with respect to the number of identities enrolled in the system. At right: comparison between the expected turnaround times t_i (indexing), t_e (exhaustive searches) in four typical scenarios.

D. Performance Optimization

Two parameters affect the performance of the proposed method: 1) the type of mother wavelet; and 2) the shape of the $w()$ sigmoid function (17) that determines the cuts in residuals per level. Figure 7 illustrates the variations in performance with respect to these choices: the upper plot expresses the results for different mother wavelets (Haar, Daubechies 2, 4,

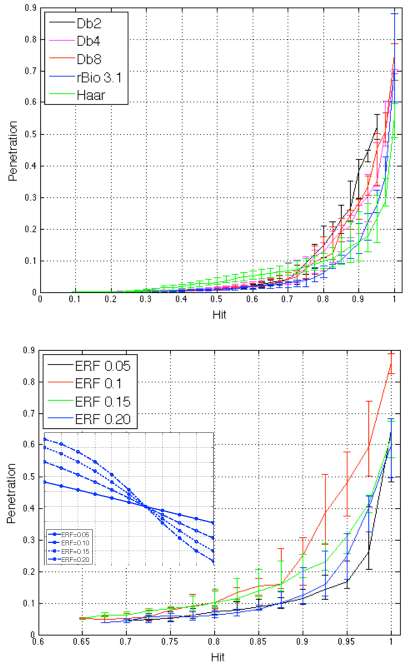


Fig. 7. Effect of the type of mother wavelet used in the decomposition / reconstruction (upper image) and of the shape of the *erf* weight function used in the retrieval phase (bottom image).

and 8 and Bi-orthogonal 3.1). The simplest wavelet (Haar) had the best performance, which was expected because *IrisCodes* are binary and maximally correlate to this type of wavelet. The bottom image gives the results for different shapes of the $w()$ function: the best performance was observed for low α values, corresponding to heavily nonlinear sigmoid shapes. Additionally, as illustrated in Figure 8, the parameter ν determines the proportion of elements stored in each level of the tree. When $\nu = 0$, all identities are stored in leaves (black bar), and as ν increases, a higher proportion of elements are stored in the upper levels of the tree. $\nu \approx 0.1N$ gave the best results in our experiments.

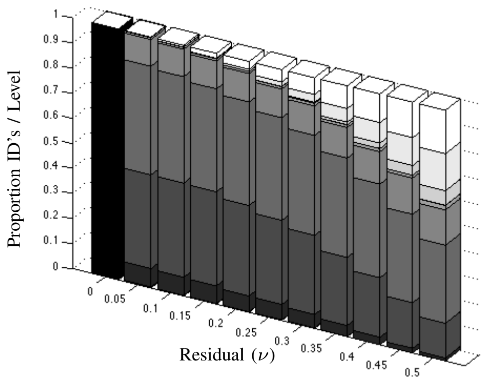


Fig. 8. Effect of the parameter ν in the proportion of elements that are stored at each level of the n-ary tree. The brightness in the bars denotes the proportion of elements per level, being elements in the deepest levels of the tree represented by the lowest intensities.

IV. Comparison With Related Methods

Performance comparison was carried out at three different levels: 1) a set of synthetic signatures was generated to perceive the effect of slight changes in genuine / impostors separability, which will be extremely hard to obtain using real data; 2) a data set of relatively well separated near-infrared data (CASIA.v4 Thousand) was used to predict the performance on scenarios that correspond to the currently deployed iris recognition systems; and 3) a data set of visible wavelength data with poor separability between classes was used (UBIRIS.v2), which fits closely the purposes of the proposed method. Three methods were selected for comparison, based on their property of operate at the *IrisCode* level: Gadde *et* al. [5], Hao *et* al. [6] and Mukherjee and Ross [13]. All the results correspond to our implementations of these algorithms.

To summarize performance by a single value, the measure due to Gadde *et* al. [5] was used, combining the hit and penetration rates. Similarly, a new measure τ was defined, corresponding to the Euclidean distance between an operating point (h, p) and the optimal performance $(h = 1, p \approx 0)$:

$$\begin{aligned} \gamma(h,p) &= \sqrt{h(1-p)} \\ \tau(h,p) &= \sqrt{(h-1)^2 + p^2}, \end{aligned} \tag{20}$$

where (h, p) express the hit and penetration rates.

A. *Synthetic* IrisCodes

A set of synthetic binary signatures was generated as described in[1]. This method is based in data correlation and simulates signatures extracted from data with varying quality, ranging from extremely degraded to optimal. This is illustrated in Figure 9, showing five decision environments, from optimal quality (Env. A, quality index=1.0) to extremely poor separated (Env. E, quality index=0.0).

When applied to good quality data, the effectiveness of the Hao *et* al.'s [6] method is remarkable (upper plot of Figure 10): this method outperforms any other by more than one order of magnitude. However, its effectiveness decreases in the case of degraded codes (bottom plot), which might be due to the concept of *multiple collisions* that becomes less effective as the probability for a collision approaches for genuine and impostor comparisons. The method of Gadde *et* al. [5] had the poorest performance for all the environments, whereas the method of Mukherjee and Ross [13] ranked third for environments of good quality. However, this method was the unique where hit values above 0.9 were not observed, neither for good quality nor degraded data.

The proposed method ranked second on good quality data and showed the least decreases in performance for degraded data. Its higher robustness was particularly evident for high hit rates, which are exactly the most important for biometrics. Table II summarizes the performance indicators (20) and the corresponding 95% confidence intervals for three types of environments. Each cell contains two values: the top value regards the full operating range, and the bottom values regard the hit $\geq$ 0.95 range. Again, these values confirm the above

[1] http://www.di.ubi.pt/~hugomcp/doc/TR_VWII.pdf

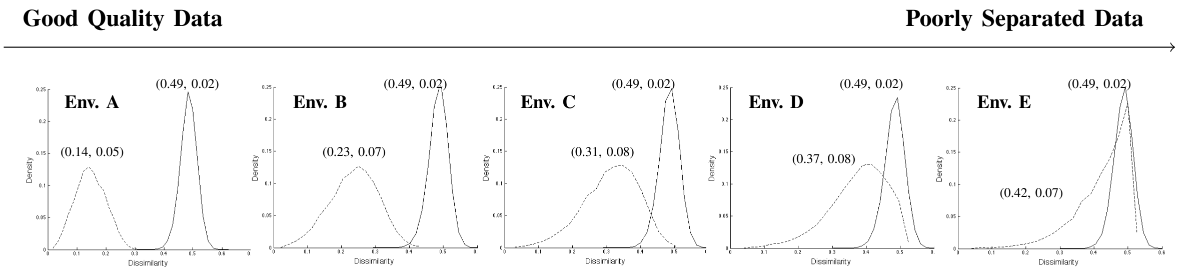


Fig. 9. Illustration of the separation between genuine (dashed lines) and impostor (continuous lines) comparisons, for different levels of *quality*. At the far left, histograms correspond to data acquired in heavily controlled scenarios (Env. A). The separability between classes decreases in the right direction.

observations about the relative performance of the methods analyzed.

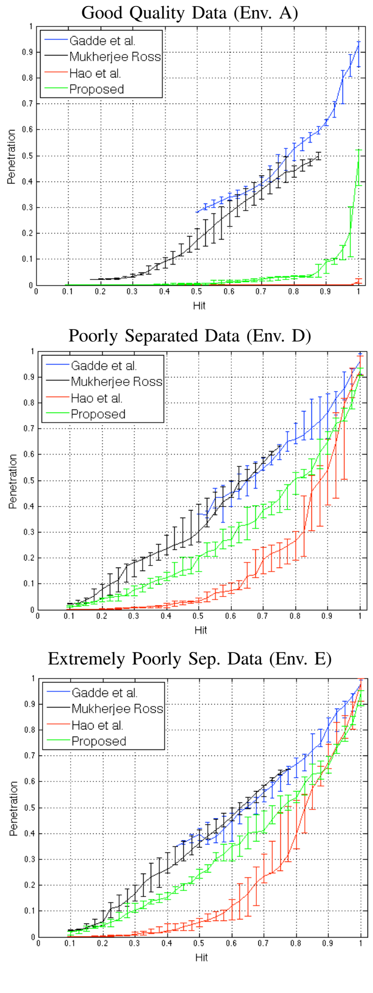


Fig. 10. Comparison between the hit / penetration rates observed for the proposed strategy and three methods used as reference. Results are expressed for three levels of data quality.

Figure 11 shows a statistic of the penetration rates (vertical axes) for queries that returned the true identity, in five environments, ranging from poorly separated (Env. quality 0.0, leftmost boxes) to good quality data (Env. quality 1.0, rightmost boxes). This plot emphasizes the extreme performance of the Hao *et* al.'s method for good quality data, having obtained penetration values close to 0. For data of reduced quality, though the median penetration value of our method is higher than Hao *et* al.'s (≈ 0.52 versus 0.13), it should be stressed that this statistic only accounts for cases in which the true identity was returned, which is more frequent in our proposal than in any other. Additionally, the inter-quartile range of penetration values was narrower in our method than in Hao *et* al.'s, which points for its highest stability in performance with respect to different queries. For all methods tested, the penetration values decreased substantially for good quality data, though this is less evident for Mukherjee and Ross' proposal. This was explained by properties of the clustering process involved here: clusters tend to have similar number of elements, and for any query, all identities inside a cluster are returned. This prevents that only a small set of identities is returned, which should happen in high quality data.

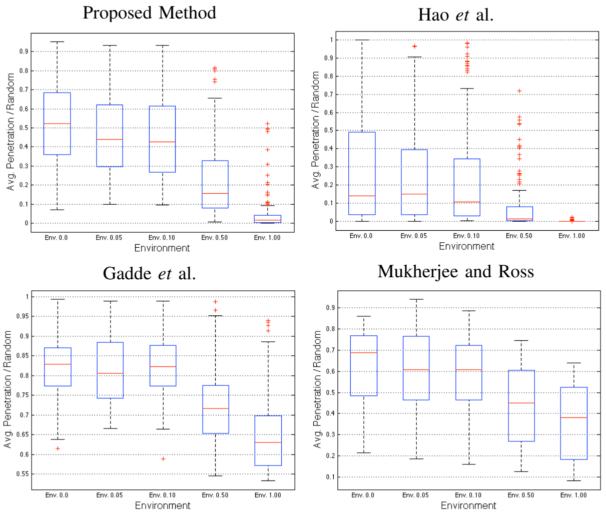


Fig. 11. Boxplots of the penetration rates observed in cases where the true identity was retrieved. Values are shown for different levels of data separability, starting from data of poorest quality (Env. 0.0) to good quality data (Env. 1.0).

To highlight the performance improvements of our proposal, Figure 12 shows a zoom-in of the hit / penetration rates for the most degraded environments. It is evident that the

TABLE II
SUMMARY OF THE PERFORMANCE INDICATORS (20) OBSERVED IN SYNTHETIC DATA, WITH RESPECT TO FOUR OTHER STRATEGIES USED AS COMPARISON TERM. THE CORRESPONDING 95% CONFIDENCE INTERVALS ARE GIVEN.

	Good Quality Data (Env. A)		Poorly Sep. Data (Env. D)		Extrem. Poorly Sep. Data (Env. E)	
Method	γ	τ	γ	τ	γ	τ
Proposed	0.91 ± 0.01 0.90 ± 0.01	0.12 ± 0.01 0.15 ± 0.01	0.67 ± 0.02 0.50 ± 0.03	0.47 ± 0.01 0.54 ± 0.02	0.64 ± 0.02 0.46 ± 0.03	0.50 ± 0.03 0.78 ± 0.01
Hao *et* al. [6]	0.99 ± 0.00 0.99 ± 0.00	0.01 ± 0.00 0.01 ± 0.00	0.76 ± 0.03 0.44 ± 0.13	0.33 ± 0.01 0.79 ± 0.13	0.74 ± 0.03 0.44 ± 0.05	0.37 ± 0.05 0.79 ± 0.02
Gadde *et* al. [5]	0.65 ± 0.01 0.44 ± 0.07	0.49 ± 0.00 0.80 ± 0.04	0.58 ± 0.03 0.37 ± 0.07	0.59 ± 0.02 0.86 ± 0.01	0.58 ± 0.02 0.31 ± 0.05	0.59 ± 0.01 0.90 ± 0.02
Mukherjee and Ross [13]	0.67 ± 0.01 -	0.48 ± 0.00 -	0.59 ± 0.03 -	0.58 ± 0.03 -	0.57 ± 0.01 -	0.60 ± 0.01 -

proposed method consistently outperforms all the others for hit values above 0.95. Additionally, it is the unique with full hit at penetration rates smaller than one (note the upper right corner of each plot), meaning that it was the unique that always retrieved the true identity *and* simultaneously reduced the search space.

The minimum hit value above which the proposed method starts to be the *best* appears to be a function of the data quality. This is evident in the bottom-right plot of Figure 12, which relates the quality of data and that minimum hit value. For the worst kind of data (Env. E), the proposed method outperforms any other for hit values above 0.88. As data quality increases, the minimum hit value varies roughly linearly, and for environments with moderate quality, the method of Hao *et* al. starts to be the best and should be used instead of ours.

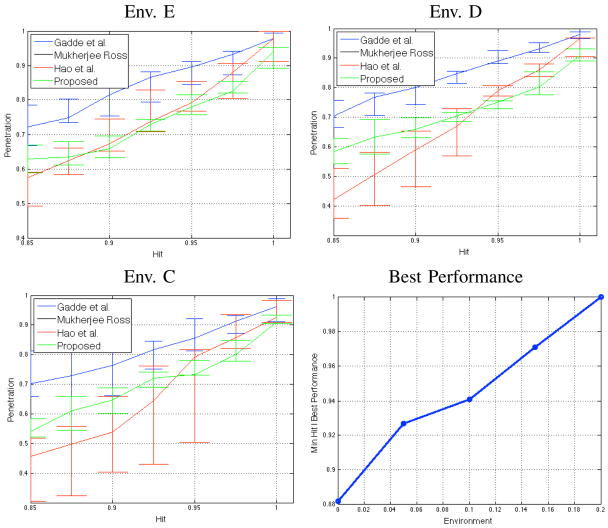


Fig. 12. Comparison between the hit/penetration plots in the performance range that was considered most important for biometric recognition purposes (hit values above 0.85). In poorly separable data the proposed method outperforms all the others, and the minimal hit value above which it becomes the best varies roughly linearly with respect to the data separability (bottom right plot).

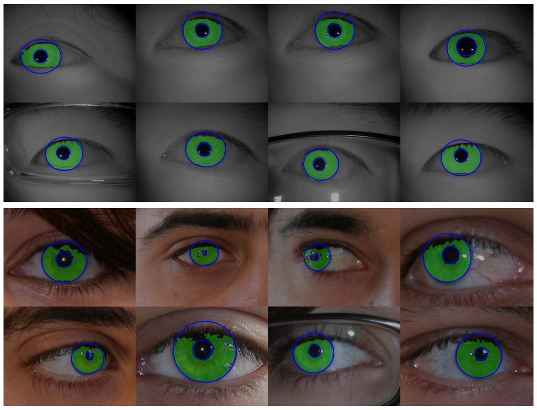


Fig. 13. Examples of the real iris images used in performance evaluation, segmented according to the method of He *et* al. [7]. The upper rows regard the CASIA.v4 Thousand data set, and the bottom rows give images of the UBIRIS.v2 data set.

B. Well Separated Near-Infrared Data

The CASIA-Iris-Thousand[2] was used to represent well separated data. It contains 20,000 images from both eyes of 1,000 subjects, enabling the evaluation with 2,000 classes.

The noise-free regions of the irises were segmented according to the method of He *et* al. [7] and an elliptical parameterization was chosen for both iris boundaries, using the random elliptic Hough transform algorithm. Next, the reasonability of the segmentation was manually verified, 110 images were discarded due to bad quality and the remaining data translated into a pseudo-polar coordinate system, using the Daugman's *rubber sheet* model. Three different configurations of Gabor kernels ***Gb*** were used in signature encoding (parameters wavelength ω and orientation θ were varied, phase ϕ and ratio r were kept constant) . The parameters for the Gabor kernels were obtained by maximizing the decidability index $d' = \frac{|\mu_I - \mu_G|}{\sqrt{\frac{1}{2}(\sigma_G^2 + \sigma_I^2)}}$, being μ_G, μ_I the means of the genuine and impostors distributions and σ_G, σ_I their standard deviations.

$$\boldsymbol{Gb}(x, y, \omega, \theta, \sigma_x, \sigma_y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}\left(\frac{\Phi_1^2}{\sigma_x^2} + \frac{\Phi_2^2}{\sigma_y^2}\right)} e^{i\frac{2\pi\Phi_1}{\omega}}, \quad (21)$$

being $\Phi_1 = x\cos(\theta) - y\sin(\theta)$, $\Phi_2 = y\sin(\theta) - x\cos(\theta)$, ω the wavelength, θ the orientation and $\sigma_x = \sigma_y = \omega/2$. The optimal parameters were found by exhaustive evaluation in a training set of 200 images randomly sampled from the initial set: $(\omega, \theta) = \{(0.33, \pi/4), (0.28, 3\pi/4), (0.51, \pi/2)\}$. Figure 13 gives some examples of the noise-free iris masks

[2] CASIA Iris Image Database: http://biometrics.idealtest.org/

and of the iris boundaries for the CASIA.v4 Thousand images.

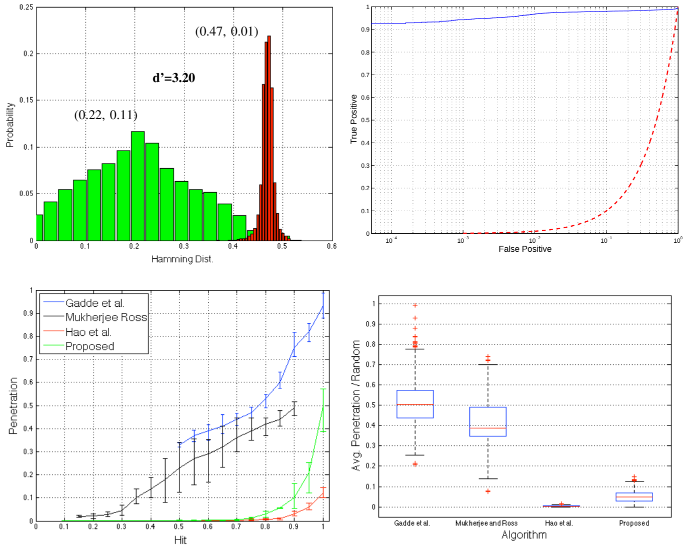


Fig. 14. Results observed for the CASIA.v4 Thousand iris data set. Plots at the upper row give the decision environment and recognition performance (in terms of ROC curves). The bottom-left plot compares the hit / penetration rates and the bottom-right plot summarizes the penetration rates observed in cases where the true identity was retrieved.

Results are given in Figure 14. Plots in the upper row give the decision environment and the recognition performance according to the feature encoding / matching algorithm used. The ROC is expressed in log scale, due to the high performance observed. In the bottom-left plot, a comparison between the hit / penetration values for the four techniques is shown, and the bottom-right plot summarizes the penetration rates in cases where the true identity was retrieved. Results accord with the previously obtained for synthetic data: the approach of Hao *et* al. largely outperformed any other. The proposed method got a consistent second rank, followed by Mukherjee and Ross' and Gadde *et* al.'s methods. The smaller variance of the proposed method and Hao *et* al.'s in the number of retrieved identities is also evident, when compared to Gadde *et* al.'s and Mukherjee and Ross'.

C. Poorly Separated Visible Wavelength Data

The UBIRIS.v2 [14] data set constitutes the largest amount of iris data acquired from large distances (four to eight meters) at visible wavelengths, containing images of poor quality that decrease the separability between the genuine / impostors matching scores. It has 11,102 images from 522 classes, from which 285 images were not considered due to their extreme bad quality (e.g., out of iris or almost completely occluded data). Similarly to the process described for the CASIA.v4 Thousand set, images were segmented according to the method of He *et* al. [7] and followed the same processing chain, yielding the Gabor filters $\boldsymbol{Gb}$ with parameters $(\omega, \theta) = \{(0.18, \pi/6), (0.35, 4\pi/6), (0.20, 7\pi/8)\}$. The bottom rows of Figure 13 illustrate some examples of the images used.

Results in this data set were regarded in a particularly positive way, as they correspond to the environments for which the proposed method was designed. As illustrated by the decision environment in the upper-left plot of Figure 15, classes have poor separability, which is confirmed by the ROC curve in the upper-right plot. For this kind of data, the proposed method outperformed all the others in the most important performance range, i.e, for hit values above 0.9 (bottom-left plot). The bottom-right plot gives a complementary perspective of the results, comparing the penetration rates in queries where the true identity was retrieved. In this case, the proposed method got higher penetration rates than Hao *et* al's, but the value for the upper whisker is particularly important: for all queries the proposed method reduced the set of identities retrieved, which did not happen in any of the remaining methods. Confirming the previous results, the method of Hao *et* al. was the best for low hit values and got a solid second place in the remaining performance range. Also, the smaller interquartile range of our method when compared to Hao *et* al.' s was also positively regarded as an indicator of its smaller variability with respect to different queries. Mukherjee and Ross' got slightly better results than Gadde *et* al.'s, but in the former method no hit values above 0.9 were observed.

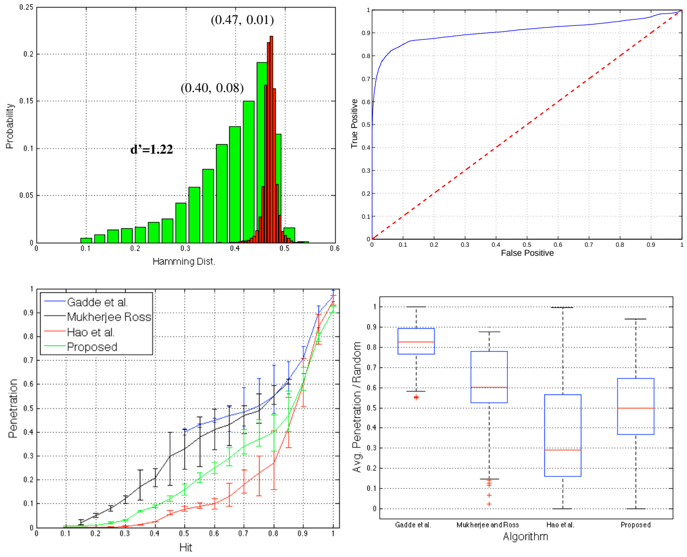


Fig. 15. Results observed for the UBIRIS.v2 iris data set. Plots at the upper row give the decision environment and recognition performance (in terms of ROC curves). The bottom left plot compares the hit / penetration rates and the bottom right plot summarizes the penetration rates observed in cases where the true identity was retrieved.

Table III summarizes the results observed in the CASIA.v4 Thousand and UBIRIS.v2 data sets. The upper value in each cell regards the full operating range and the bottom value regards the range most important for biometrics (hit values above 0.95). The values highlighted in bold confirm the suitability of the proposed method to work on low quality data (UBIRIS.v2, $\Delta\gamma = +0.11$ for our method, when compared to Hao *et* al.'s) and stress the effectiveness of Hao *et* al.'s method to work in scenarios that correspond to the currently deployed iris recognition systems (CASIA.v4 Thousand, $\Delta\gamma = -0.07$ for our method, with respect to Hao *et* al.'s).

TABLE III
SUMMARY OF THE PERFORMANCE INDICATORS (20) OBSERVED IN THE CASIA.V4 THOUSAND AND UBIRIS.V2 DATA SETS, WITH RESPECT TO FOUR STRATEGIES USED AS COMPARISON TERMS. THE CORRESPONDING 95% CONFIDENCE INTERVALS ARE GIVEN.

	CASIA.v4 Thousand (NIR)		UBIRIS.v2 (VW)	
Method	γ	τ	γ	τ
Proposed	0.91 ± 0.02 **0.88 ± 0.02**	0.12 ± 0.01 **0.14 ± 0.02**	0.71 ± 0.02 **0.53 ± 0.03**	0.36 ± 0.02 **0.78 ± 0.02**
Hao *et* al. [6]	0.96 ± 0.01 **0.95 ± 0.01**	0.04 ± 0.01 **0.05 ± 0.01**	0.75 ± 0.03 **0.42 ± 0.06**	0.34 ± 0.02 **0.82 ± 0.04**
Gadde *et* al. [5]	0.62 ± 0.01 0.40 ± 0.07	0.51 ± 0.02 0.82 ± 0.02	0.60 ± 0.02 0.37 ± 0.04	0.47 ± 0.02 0.88 ± 0.03
Mukherjee and Ross [13]	0.76 ± 0.02 -	0.43 ± 0.02 -	0.61 ± 0.02 -	0.46 ± 0.02 -

V. CONCLUSIONS

This paper proposed an indexing / retrieval method to operate in *IrisCodes* extracted from low quality data [3], i.e., with a poor separability between the genuine and impostor matching scores. The proposed strategy is based on the decomposition of *Iriscodes* at multiple scales and in their placement in nodes of an n-ary tree. In retrieval, only a few paths in the tree are traversed before the stopping criterion is achieved. The main contributions are three-fold: 1) the proposed method has consistent advantages over other techniques when applied to low quality data. This is particularly evident in the performance range that is most important for biometrics (hit values above 0.95); 2) these levels of performance were obtained at a reduced computational cost, making the proposed method advantageous (compared to sequential searches in terms of turnaround time) when more than 54,000 identities are enrolled in the system; and 3) the method is compatible with different iris signature encoding schemes, provided that they produce a binary signature, and use wavelets that resemble (at least roughly) the ones used in indexing.

ACKNOWLEDGEMENTS

The financial support given by ”FCT-Fundação para a Ciência e Tecnologia” and ”FEDER” in the scope of the PTDC/EIA/103945/2008 research project ”NECOVID:Negative Covert Biometric Recognition” is acknowledged. Portions of the research in this paper use the CASIA-IrisV4 collected by the Chinese Academy of Sciences’ Institute of Automation (CASIA).

REFERENCES

[1] J. Daugman. Probing the uniqueness and randomness of IrisCodes: Results from 200 billion iris pair comparisons. *Proceedings of the IEEE*, vol. 94, no. 11, pag. 1927-1935, 2006.

[2] Emirates ID Head Office. Iris scan prevents entry of 350,000 deportees: Saif Bin Zayed. http://www.emiratesid.gov.ae/en/media-centre/news/, assessed on June 2013.

[3] D. Donoho and I. Johnstone. Ideal Spatial Adaptation by Wavelet Shrinkage. *Biometrika*, vol. 81, pag. 425-455, 1994.

[4] J. Fu, H. Caulfield, S. Yoo and V. Atluri. Use of Artificial Color filtering to improve iris recognition and searching. *Pattern Recognition Letters*, vol. 26, pag. 2244-2251, 2005.

[5] R. Gadde, D. Adjeroh and A. Ross. Indexing Iris Images Using the Burrows-Wheeler Transform. *Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS)*, pag. 1-6, 2010.

[6] F. Hao, J. Daugman and P. Zielinski. A Fast Search Algorithm for a Large Fuzzy Database. *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 2, pag. 203-211, 2008.

[7] Z. He, T. Tan, Z. Sun and X. Qiu Towards Accurate and Fast Iris Segmentation for Iris Biometrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 9, pag. 1617-1632, 2009.

[8] Institute of Automation, Chinese Academy of Sciences. CASIA Iris Image Database. http://www.cbsr.ia.ac.cn/IrisDatabase, 2009.

[9] U. Jayaraman and S. Prakash. An Iris Retrieval Technique Based on Color and Texture. *Proceedings of the Seventh Indian Conference on Computer Vision, Graphics and Image Processing*, pag. 93-100, 2010.

[10] S. Mallat. A Wavelet Tour of Signal Processing. *Academic Press*, ISBN: 0-12-466606-X, 1999.

[11] H. Mehrotra, B. Majhi and P. Gupta. Robust iris indexing scheme using geometric hashing of SIFT keypoints. *Journal of Network and Computer Applications*, vol. 33, pag. 300-313, 2010.

[12] H. Mehrotra, B. Srinivas, B. Majhi and P. Gupta. Indexing Iris Biometric Database Using Energy Histogram of DCT Subbands. *Journal of Communications in Computer and Information Science*, vol. 40, pag. 194-204, 2009.

[13] R. Mukherjee and A. Ross. Indexing Iris Images. *Proceedings of the 19th International Conference on Pattern Recognition (ICPR 2008)*, pag. 1-4, 2008.

[14] H. Proença, S. Filipe, R. Santos, J. Oliveira and L. A. Alexandre. The UBIRIS.v2: A Database of Visible Wavelength Iris Images Captured On-The-Move and At-A-Distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 8, pages 1502–1516, 2010.

[15] N. Puhan and N. Sudha. Coarse indexing of iris database based on iris color. *International Journal on Biometrics*, vol. 3, no. 4, pag. 353-375, 2011.

[16] X. Qiu, Z. Sun and T. Tan. Coarse Iris Classification by Learned Visual Dictionary. *Proceedings of the International Conference on Biometrics, Lecture Notes on Computer Science*, vol. 4642, pag. 770-779, 2007.

[17] Unique Identification Authority of India. [online], http://uidai.gov.in/about-uidai.html, accessed on June, 2013.

[18] M. Vatsa, R. Singh and A. Noore. Improving Iris Recognition Performance Using Segmentation, Quality Enhancement, Match Score Fusion, and Indexing. *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics*, vol. 38, no. 4, pag. 1021-1035, 2008.

[19] L. Yu, K. Wang and D. Zhang. A Novel Method for Coarse Iris Classification. *Proceedings of the International Conference on Biometrics, Lecture Notes on Computer Science*, vol. 3832, pag. 404-410, 2006.

[20] Q. Zhao. A New Approach for Noisy Iris Database Indexing Based on Color Information. *Proceedings of the 6th International Conference on Computer Science and Education (ICCSE 2011)*, pag. 28-31, 2011.

Hugo Proença received the B.Sc. degree from the University of Beira Interior, Portugal, in 2001, the M.Sc. degree from the Faculty of Engineering, University of Oporto, in 2004, and the Ph.D. degree from the University of Beira Interior, in 2007. His research interests are focused in the artificial intelligence, pattern recognition and biometrics. Currently, he serves as Assistant Professor in the Department of Computer Science, University of Beira Interior. He is the area editor (ocular biometrics) of the IEEE Biometrics Compendium Journal and member of the Editorial Board of the International Journal of Biometrics. Also, he served as Guest Editor of special issues of the Pattern Recognition Letters, Image and Vision Computing and Signal, Image and Video Processing journals.

[3] All the materials and information required to reproduce the results given in this paper are available at: http://www.di.ubi.pt/∼hugomcp/VWII.