Iris Biometric Indexing

H. Proença and J. C. Neves,

Abstract Indexing / retrieving sets of iris biometric signatures has been a topic of increasing popularity, mostly due to the deployment of iris recognition systems in nationwide scale scenarios. In these conditions, for each identification attempt, there might exist hundreds of millions of enrolled identities and is unrealistic to match the probe against all gallery elements in a reasonable amount of time. Hence, the idea of indexing / retrieval is - upon receiving one sample - to find in a quick way a sub-set of elements in the database that most probably contains the identity of interest, i.e., the one corresponding to the probe. Most of the state-of-the-art strategies to index iris biometric signatures were devised to decision environments with a clear separation between genuine and impostor matching scores. However, if iris recognition systems work in low quality data, the resulting decision environments are poorly separable, with a significant overlap between the distributions of both matching scores. This chapter summarises the state-of-the-art in terms of iris biometric indexing / retrieval and focuses in an indexing / retrieval method for such low quality data and operates at the *code* level, i.e., after the signature encoding process. Gallery codes are decomposed at multiple scales, and using the most reliable components of each scale, their position in a n-ary tree is determined. During 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 branches is traversed up to the last level.

1 Introduction

Iris biometrics is now used in various scenarios with satisfactory results (e.g., refugee control, security assessments and forensics) and nationwide deployment of iris recognition systems has already begun. In the last information update about the

University of Beira Interior, Department of Computer Science, IT- Instituto de Telecomunicações, 6201-001 Covilhã, Portugal. e-mail: {hugomcp,jcneves}@di.ubi.pt

UAE system [3], the number of enrolled identities was over 800,000, and more than $2e^{12}$ iris cross-comparisons have been performed. The Unique Identification Authority of India [21] is developing the largest-scale recognition system in the world (over 1 200 million persons). Similarly, the United Kingdom ID card initiative [22] intends to provide one biometric identity for each citizen, which will result in 90 million enrolled identities if the goals are fully met.

Though matching *IrisCodes* primarily involves the accumulation of bitwise XOR operations on binary sequences, an increase in turnaround time occurs in national or continental contexts, which motivated growing interest in iris indexing strategies able to reduce the turnaround time without substantially affecting precision. As noted by Hao *et* al. [7], the indexing problem is a specific case of the more general nearest neighbor search problem, and motivated several proposals in the last few years (section 2). However, most of these methods were devised to decisions environments of good quality, with a clear separation between the matching scores of genuine and impostors comparisons.

In this chapter, not only we summarise the state-of-the-art in terms of iris biometric indexing / retrieval, but we focus particularly the problem of indexing in decisions environments of poor quality, with a significant overlap between the matching scores of genuine and impostor comparisons. This kind of environments is likely when iris recognition systems operate in non-controlled data acquisition protocols (e.g., automated surveillance scenarios, using COTS hardware). We analyse a method [17] that operates at the code level, i.e., after the *IrisCode* encoding process. We decompose the codes at multiple levels and find their most reliable components, determining their position in an n-ary tree. During retrieval, the probe is decomposed similarly, and distances to the multi-scale centroids are obtained, penalizing paths of the tree and traversing a only a subset up to the leaves.

The remainder of this paper is organized as follows: Section 2 summarizes the state-of-the-art in terms of the published indexing / retrieval strategies. Section 3 provides a description of the method we focus in the chapter. Section 4 presents and discusses the results with respect to state-of-the-art techniques. Finally, the conclusions are given in Section 5.

2 State-of-the-Art

Table 1 summarizes the iris indexing methods that were reported recently in the literature, which can be coarsely classified using two criteria: 1) the light spectrum used for the data acquisition (either at near-infrared or visible wavelengths); and 2) the methods' input, which is either the raw iris texture or the biometric signature (*IrisCode*).

Table 1 Overview of the most relevant recently published iris indexing methods. *NIR* stands for near-infrared and *VW* for visible wavelength data.

Method	**Type**	**Spectrum**	**Preprocessing**	**Summary**
Fu *et* al. [5]	Color	Own (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
Gadde *et* al. [6]	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
Hao *et* al. [7]	*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
Jayaraman and Prakash [10]	Color, Texture	UBIRIS.v1, UPOL (VW)	Segmentation	Color analysis in YCbCr space. SURF keypoint description, Kd-tree indexing
Mehrotra *et* al. [13]	Texture	CASIA.1, ICE, WVU (NIR)	Segmentation)	Keypoints localization, geometric analysis, hash table construction
Mehrotra *et* al. [14]	Texture	CASIA, Bath, IITK (NIR)	Segmentation, normalization	Multi-resolution decomposition (DCT). Energy of sub-bands extraced in Morton order, B-tree indexing
Rathgeb *et* al. [20]	*IrisCode*	IITD (NIR)	Segmentation, normalization	1) sub-blocks division, Bloom filters, tree partition and traversal
Mukherjee and Ross [15]	Texture, *IrisCode*	CASIA-V3 (NIR)	Segmentation, normalization	1) sub-blocks division, top-n similarity between blocks, tree partition; 2) sublocks partition, k-means clustering
Puhan and Sudha [18]	Color	UBIRIS.v1, UPOL (VW)	Segmentation	Conversion to YCbCr, semantic decision tree
Qiu *et* al. [19]	Texture	CASIA.1, ICE, WVU (NIR)	Segmentation, normalization	Extraction of texton histograms, Chi-square dissimilarity, K-means clustering
Vatsa *et* al. [23]	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
Yu *et* al. [24]	Texture	CASIA.1, ICE, WVU (NIR)	Segmentation, normalization	Definition of radial ROIs, extraction of local fractal dimensions, semantic decision tree
Zhao [25]	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 enrolled from corresponding bins

Yu *et* al. [24] represented the normalized iris data in the polar domain, dividing it radially into sixteen regions, and obtaining the fractal dimension for each one. Using first-order statistics, a set of semantic rules indexes the data into one of four classes. During retrieval, each probe is matched exclusively against gallery data in the same class. Fu *et* al.'s [5] use color information and suggest that artificial color filters provide an orthogonal discriminator of the spatial iris patterns. Each color filter is represented by a discriminator that operates at the pixel level. Gadde *et* al. [6] analyzed the distribution of intensities and selected patterns with low coefficients of variation (CVs) as indexing pivots. For each probe represented in the polar domain, a radial division of n-bands was performed and indexed using the highest density of CV patterns. Hao *et* al. [7] exclusively relied in the *IrisCode*. Using the spatial spread of the most reliable bits, they were based on the notion of multi-collisions. In the retrieval process, a minimum of k collisions between the probe and gallery samples is required to identify a potential match. Jayaraman and Prakash [10] fused texture and color information: they estimated the iris color in the YCbCr space and determined an index to reduce the search space. Texture was encoded using SURF [1] keypoint detection and description. Mehrotra *et* al. [13] used SIFT [11] descriptors and their spatial distribution. To overcome the effect of non-uniform illumination and partial occlusions caused by eyelids, keypoints were extracted from angularly constrained regions of the iris. During retrieval, the geometric hashed location determined from the probe data accesses the appropriate bin of a hash table, and for every entry found, a vote is cast. The identities that receive more than a certain number of votes are considered possible candidates. Mehrotra *et* al. [14] divided the polar iris data into sub-bands using a multi-resolution Discrete Cosine transformation. Energy-based histograms were extracted for each band, divided into fixed-size bins, and iris images with similar energy values were grouped. A B-tree in which instances with the same key appear in the same leaf node was built. For a query, the corresponding key was generated, and the tree was traversed until a leaf node was reached. The templates stored at the leaf node were retrieved and compared with the query template to find the best match. Mukherjee and Ross [15] approached the problem from two different perspectives: by analyzing the iris texture and the *IrisCode*. The best results in the latter case were attained when each code was split into fixed-size blocks. First-order statistics for each block were used as the primary indexing value. A k-means strategy was used to divide the feature space into different classes. Qiu *et* al. [19] created a small finite dictionary of visual words (clusters in the feature space), called *textons*, to represent visual primitives of iris images. Then, texton histograms were used to represent the global features, and the k-means algorithm was used to classify the irises into five categories. Vatsa *et* al. [23] represented pixels of sub-regions of the unwrapped iris data in an 8-D binary feature space. The four most significant bits were used to build four corresponding maps from which the Euler numbers were extracted. Retrieving was performed using the nearest neighbor technique for the topological data. Zhao [25] determined the average RGB values inside the iris, weighted them by the luminance component to form a 3D feature space, and subsequently projected them into independent 1-D spaces. Probes were matched only against gallery samples that corresponded to the union

of the identities in the bins of each space. A similar approach was due to Puhan and Sudha [18]: they obtained the color index (in the YCbCr color space) and used a semantic decision tree to index the database.

3 Indexing / Retrieving Poorly Separated Data

3.1 Indexing

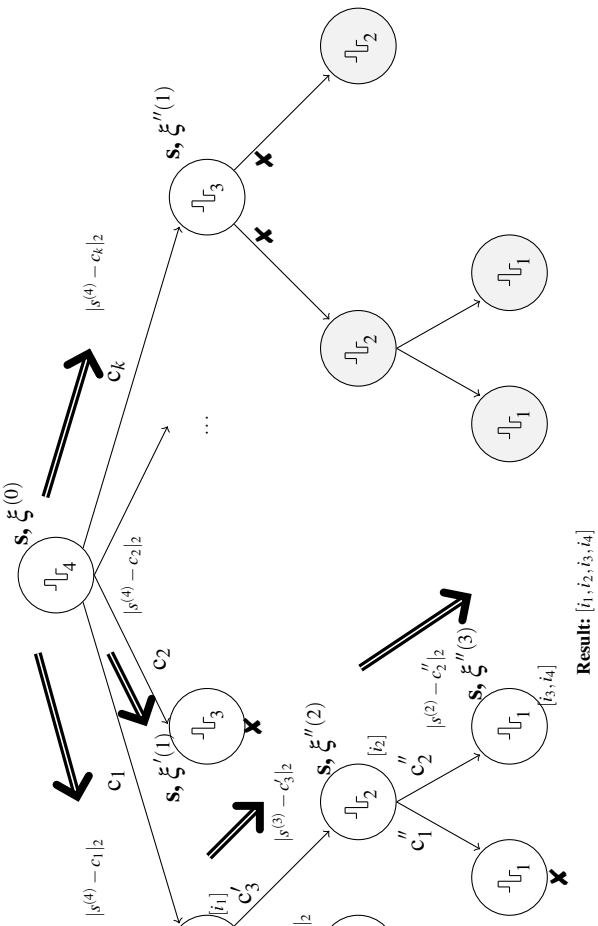

Fig. 1 Cohesive perspective of the indexing structure and of the retrieval strategy focused in this chapter. For a query s with *residual* $\xi^{(0)}$, the distance between the decomposition of s at top level ($s^{(4)}$) to the centroids is used to generate the new generation of residuals ($\xi^{(1)}$). 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 level where $\xi^{(\cdot)} > 0$ is included in the retrieved set.

3.1.1 Codes Decomposition / Reconstruction

Let s_i denote a signature (*IrisCode*) s from subject i. As illustrated in Fig. 1, the key insight of the method we describe here is to obtain coarse-to-fine representations of s_i as a function of the level l in the tree ($s_i^{(l)}$). These representations are grouped according to their similarity in the L_2 metric space, and stored in tree nodes. A node is considered a leaf when a cluster centroid is at a sufficiently small distance from $s_j^{(l)}$, $\forall j$.

Let $\phi(x) = \sum_{k\in\mathbb{Z}} h(k)\sqrt{2}\phi(2x-k)$ and $\psi(x) = \sum_{k\in\mathbb{Z}} g(k)\sqrt{2}\phi(2x-k)$ be two filters, where $h(.)$ and $g(.)$ are low-pass and high-pass filters. According to Mallat's multiresolution analysis [12], the operator representation of these filters is defined by

$$(H_a)_k = \sum_n h(n-2k)a_n$$
$$(G_a)_k = \sum_n g(n-2k)a_n,$$

where H and G correspond to one-step wavelet decomposition. Let $s^{(n)}$ denote the original signal of length $N = 2^n$ (in our experiments, $n = 11$). $s^{(n)}$ is represented by a linear combination of ϕ filters:

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

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

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

where $\left(\sum_{i=0}^{n-1} len(d^{(i)})\right) + len(s^{(0)}) = len(s^{(n)}) = 2^n$; $len(.)$ is the number of signal coefficients. $s^{(n)}$ are approximated at different levels l using H^* and G^* reconstruction filters:

$$(H_a^*)_l = \sum_k h(l-2k)a_k$$
$$(G_a^*)_l = \sum_k g(l-2k)a_k,$$

where $s^{(n)} := \sum_{j=0}^{n-1} (H^*)^{(j)} G^* d^{(j)} + (H^*)^{(n)} G^* s^{(0)}$. Considering that *IrisCodes* are binary, the Haar wavelet maximally correlates them and was considered the most

convenient for this purpose: the filter coefficients are given by $h = [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]$, $g = [\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}]$ and the reconstruction coefficients are similar $h^* = h$ and $g^* = -g$. Under this decomposition strategy, H acts as a smoothing filter and G as a detail filter. Next, the filters are combined to reconstruct the signal at multiple levels by removing the small-magnitude detail coefficients that intuitively do not significantly affect the original signal. This is possible because the wavelets provide an unconditional basis, i.e., one can determine whether an element belongs to the space by analyzing the magnitudes of the coefficients used in the linear combination of the basis vectors.

The adjustment of a threshold (λ) for the minimal magnitude of the coefficients used in the reconstruction was accomplished according to the idea of universal threshold, proposed by Donoho and Johnstone [4]. Here, wavelet coefficients with a magnitude smaller than the expected maximum for an independent and identically distributed noise sequence that is normally distributed were ignored:

$$\lambda = \sqrt{2log(n)}\hat{\sigma}, \tag{1}$$

where 2^n is the length of the original signal and σ is estimated using

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

where $d_{i,j}$ denotes the ith wavelet coefficient at level j and $\bar{d}$ is the mean of coefficients. Figure 2 illustrates an *IrisCode* s_i and its representations at different levels ($n = \{1,2,10\}$). The coarsest representation $s^{(10)}$ retains the lowest frequency components of the input code (intensities are stretched for visualization purposes) and is used in the root of the indexing tree, whereas the finest representation $s^{(1)}$ is used at the leaves.

As illustrated in Figure 3, $s^{(i)}$ correspond to increasingly smoothed versions of s. They were used at each level of the n-ary tree, starting from the coarsest reconstruction (root of the tree) and iteratively adding detail coefficients at the deeper levels. The top plot shows the average residuals between the original signal and the reconstruction with respect to the levels used (horizontal axis); being evident that - on average - residuals decrease directly with respect to the decomposition level. The plots at the bottom row show histograms of the residuals for the coarsest (center) and finest scales (right); enabling to perceive that the coarsest-scale reconstruction is essentially a mean of the original signal.

3.1.2 Determining the Number of Branches per Node

Having a set of reconstructed signals $\{s_i^{(l)}\}$, a clustering algorithm was used to find centroid that corresponds to a node in the tree and a partition of $\{s_i^{(l)}\}$, according to the distances of elements to that centroid. Also, if the distance between $\{s_i^{(l)}\}$ and

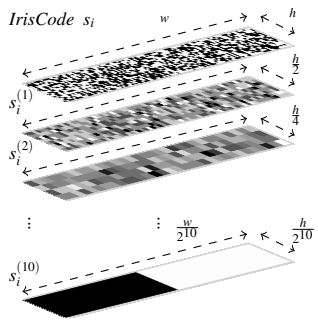

Fig. 2 Representation of an *IrisCode* (upper image) at different levels, retaining coarse (bottom image) to fine information from an input code. The $s_i^{(10)}$ representation is used in the root of the indexing tree and the remaining representations at the deeper levels of the tree. Intensities and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

the cluster centroid is less than a residual ($v \approx n \cdot 0.1, \forall i$), the indexing process stops at that level 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 determine the *optimal* value, a comparison between the proportion of variance in the data with respect to the number of clusters was carried out. Intuitively, if the number of clusters is too low, new partitions reduce the variance significantly, but - at the other extreme - if the number of clusters is too high, adding a new one almost doesn't reduce variance. Hence, the ideal number of clusters was considered to be reached when this marginal gain decreases significantly, Let k be the number of clusters. The proportion of the variance explained is characterized by a F-test:

$$F(k) = \frac{(n-k)\sum_{i=1}^{k} n_i(\overline{Y_i} - \overline{Y})^2}{(k-1)\sum_{i=1}^{k}\sum_{j=1}^{k}(\overline{Y_{ij}} - \overline{Y_i})^2}, \tag{3}$$

where $\overline{Y_i}$ is the sample mean in the cluster, n_i is the number of codes and $\overline{Y}$ the overall mean. Considering $(k_i, F(k_i))$ as points on a curve, we seek the value with minimal curvature, which corresponds to the number of clusters at which the marginal gain drops more. Parameterizing the curve $(x(t), y(t)) = (k_i', F(k_i'))$ using quadratic polynomials yields a polygon with segments defined by

$$\begin{cases} x(t) = a_3 t^2 + a_2 t + a_1 \\ y(t) = b_3 t^2 + b_2 t + b_1 \end{cases} \tag{4}$$

The $x(t)$ and $y(t)$ polynomials were fitted via the least squares strategy using the previous and next points at each point to find the a_i and b_i coefficients:

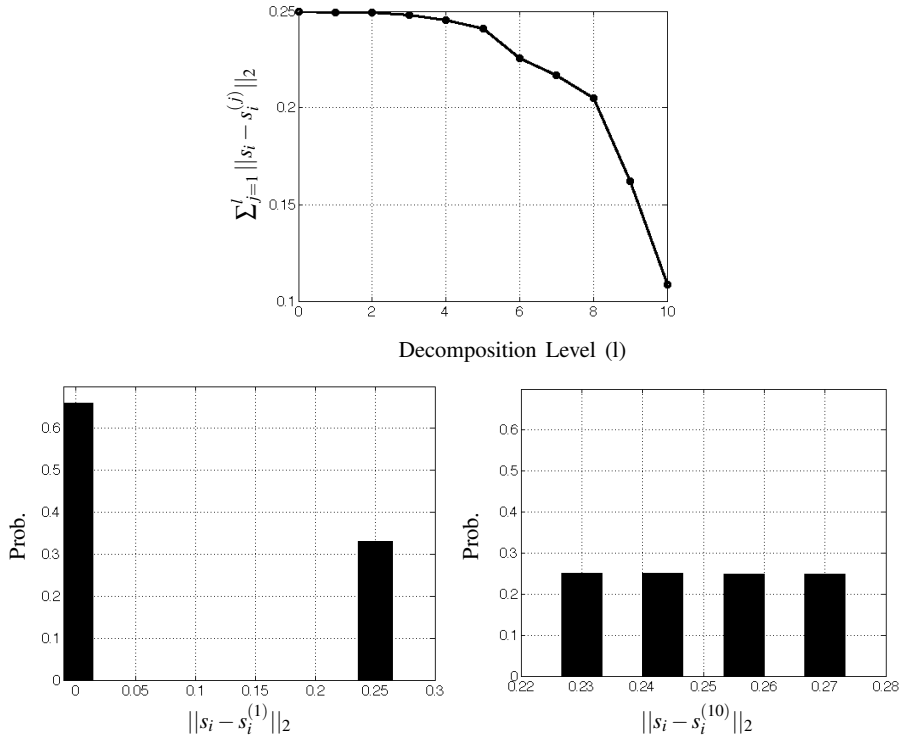

Fig. 3 Average sum of residuals between an *IrisCode* s_i and its representations at different levels ($s_i^{(l)}$, top image). The images in the bottom row give the histograms of the residuals observed for decompositions/reconstructions at the coarsest (left) and finest (right) levels.

$$\Upsilon_a = \sum_{i=1}^{3} \left[y_i - a_1 + a_2 x_i + a_3 x_i^2 \right]^2. \tag{5}$$

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

$$\begin{bmatrix} 3 & \sum x_i & \sum x_i^2 \\ \sum x_i & \sum x_i^2 & \sum x_i^3 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_i y_i \\ \sum x_i^2 y_i \end{bmatrix} \tag{6}$$

By solving the system of linear equations for a_i, the coefficients of the interpolating polynomials are obtained. The b_i values are obtained similarly. The curvature κ at each point k_i is given by

$$\kappa(k_i) = \frac{x(t)'y(t)'' - y(t)'x(t)''}{\sqrt{(x(t)'^2 + y(t)'^2)^3}}, \tag{7}$$

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, (7) was rewritten as

$$\kappa(k_i) = \frac{(2ta_3 + a_2)2b_3 - 2a_3(2b_3 + b_2)}{((2ta_3 + a_2)^2 + (2tb_3 + b_2)^2)^{\frac{3}{2}}}. \tag{8}$$

Because we are primarily interested in the curvature at each point, t was replaced by 0, yielding

$$\kappa(k_i) = \frac{2(a_2b_3 - a_3b_2)}{(a_2^2 + b_2^2)^{\frac{3}{2}}}. \tag{9}$$

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

$$\hat{k} = \arg\min_i \kappa(k_i) \tag{10}$$

Figure 4 shows an example of the described strategy to find the number of clusters per node. Here, the $F(k_i)$ values were tested for $k_i \in \{2, \ldots, 11\}$ (continuous lines). The dashed line corresponds to the $\kappa(k_i)$ values. The minimum curvature of the interpolating polynomials was observed at $\hat{k} = 8$.

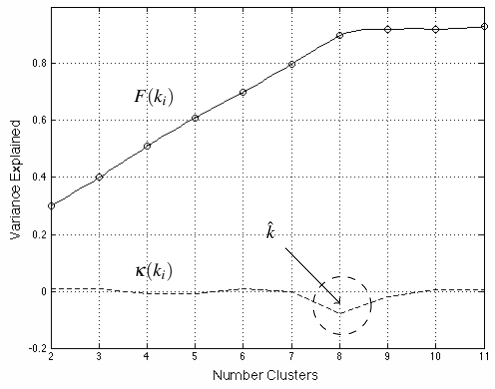

Fig. 4 Illustration of the strategy used to determine the number of clusters at each node of the n-ary tree. For ($k_i \in \{2, \ldots, 11\}$), the amount of variance explained $F(k_i)$, is denoted by circular data points. Quadratic polynomials were fitted to interpolate this data (continuous lines), 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 drops, i.e., where the curvature value attains a minimum.

3.2 Retrieval

The retrieval process receives a query signature s and a residual value $\xi > 0$. The idea is to traverse only a subset of the paths in the tree, by iteratively decreasing ξ and stopping when $\xi < 0$. At each node, the L_2 distance between the reconstructed version of the query at level (l) and a cluster centroid is subtracted from ξ, considering the maximum distance between that centroid and the identities stored in the branch. Formally, let $q(s, \xi^{(0)})$ be the query parameters at the tree root (level l). $s^{(l)}$ is the reconstruction of s at the highest scale. The next generation of residual values $\xi^{(l-1)}$ at the child nodes is given by

$$\xi^{(l-1)} = \xi^{(l)} - \max\left\{0, ||s^{(l)} - c_i^{(l)}||_2 - \max\{||s_j^{(l)} - c_i^{(l)}||_2, j \in \{1,\ldots,t_i\}\}\right\}, \quad (11)$$

being c_i the i^{th} cluster and $s_j^{(l)}$ the reconstruction at scale l of the signatures in that branch of the tree. The set of identities retrieved is obtained by

$$q(s, \xi^{(l)}) = \begin{cases} [\{i_.\}, q(s, \xi^{(l-1)})], \text{ , if } \xi^{(l)} > 0 \wedge l > 1 \\ \{i_.\} \text{ , if } \xi^{(l)} > 0 \wedge l = 1 \\ \emptyset \text{ , if } \xi^{(l)} \leq 0 \end{cases} \quad (12)$$

where $[,]$ denotes vector concatenation and $\{i_.\}$ is the set of identities in each node. Because of the intrinsic properties of wavelet decomposition / reconstruction, the distance values at the higher scales should be weighted, as they represent more signal components. This was done by the *erf* function:

$$w(l) = \frac{1 + \text{erf}(\alpha(l-n))}{2}, \quad (13)$$

being α a parameter that controls the shape of the sigmoid. Figure 5 shows examples of histograms of the cuts in residuals ξ with respect to the level in the tree. On the horizontal axis, note the decreasing magnitudes with respect to level. The dashed vertical lines indicate the residual cuts in the tree path that contained the true identity. Note that, with exception to the leaf level ($l = 1$), no cuts in the residual were performed for the paths that correspond to the true identity. This is in opposition to the remaining paths on the tree, where cuts in residual occurred at all levels.

3.3 Time complexity

Here we are primarily interested in analyzing the time complexity of the retrieving 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 segmentation, coding and matching stages. Without indexing, the average turnaround time for an exhaustive search T_e is given by

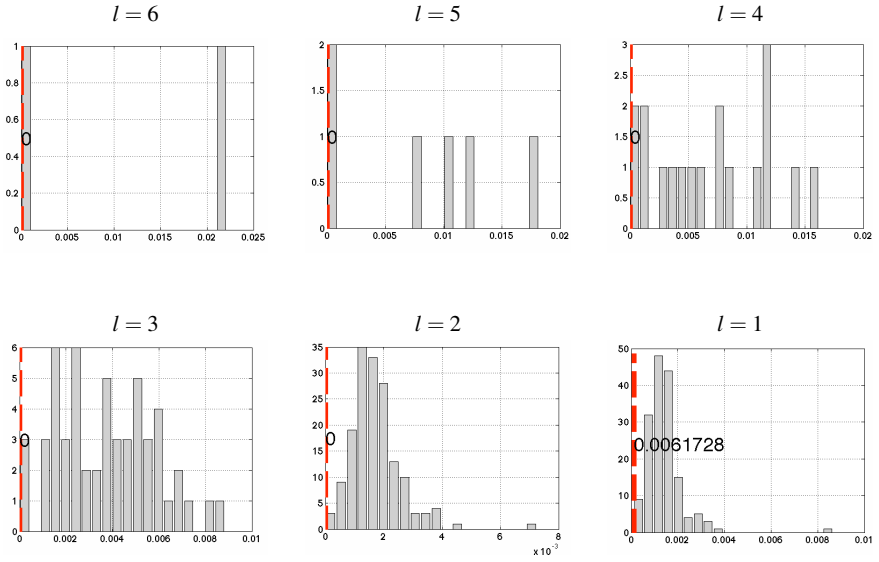

Fig. 5 Histograms of the cuts in residuals $\xi^{(l)}$ per level during retrieval. The vertical dashed lines give the cumulative distribution values of the cuts, in the paths that correspond to the matching identity of the query. Gray bars express the frequencies of the cuts occurred in the remaining paths of the tree.

$$T_e = T_s + T_c + N\ 0.5\ T_m, \tag{14}$$

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

$$T_i = T_s + T_c + N\ T_r + (h\ p + (1-h))\ 0.5\ N\ T_m, \tag{15}$$

being T_r the average turnaround time for retrieval and h and p the hit and penetration rates. Figure 6 compares the expected values for the T_i and T_e turnaround times with respect to the number of identities enrolled. 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 analysed method and of exhaustive searches. The horizontal bars near each point give the range of values observed, enabling to conclude that the indexing / retrieving 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 data quality. Even though, it serves as an approximation of the minimum number of identities that turn the indexing process advantageous in terms of turnaround time.

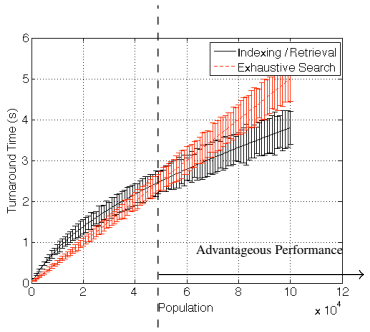

Fig. 6 Comparison between the turnaround times of an exhaustive search (red line) and when using the indexing / retrieval strategy analysed in this chapter (black line), with respect to the number of identities enrolled in the system.

4 Performance Comparison

Performance comparison was carried out at three different levels: 1) a set of synthetic signatures was generated to perceive performance with respect to slight changes in classes separability, which will be extremely hard to obtain using real data; 2) a data set of relatively well separated near infra-red data (CASIA.v4 Thousand) was used, in order to predict performance on scenarios that correspond to the currently deployed iris recognition systems; and 3) a data set of visible wavelength data with poor classes separability was used (UBIRIS.v2), which fits closely the purposes of the method described in this chapter. Four methods were selected for comparison, based on their property of operating at the *IrisCode* level: Gadde *et* al. [6], Hao *et* al. [7] and Mukherjee and Ross [15]. All the results correspond to our implementations of these techniques. In an appendix, detailed instructions to access the source code implementations are given.

To summarize performance by a single value, the proposal of Gadde *et* al. [6] was used, combining the hit and penetration rates. Similarly, a new measure τ corresponding to the Euclidean distance between an operating point (h,p) and the optimal performance$(\text{hit}=1,\text{penetration}\approx 0)$, was defined:

$$\gamma(h,p) = \sqrt{h(1-p)} \tag{16}$$

$$\tau(h,p) = \sqrt{(h-1)^2+p^2}, \tag{17}$$

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

4.1 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 broad levels of quality, ranging from extremely degraded to optimal. This is illustrated in Figure 7, showing various decision environments, from optimal (Env. A) to extremely poor separated (Env. C).

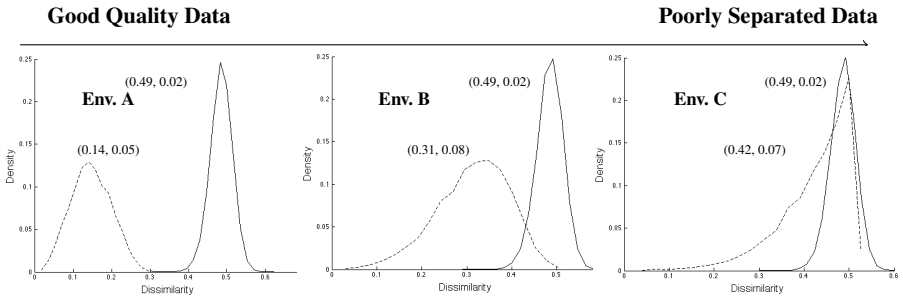

Fig. 7 Illustration of the separation between genuine (dashed lines) and impostor (continuous lines) comparisons, for different levels of *quality*. At the far left, histograms corresponding to data acquired in heavily controlled scenarios are shown (A). Classes separability decreases in the right direction.

When applied to good-quality data, the effectiveness of the Hao *et* al. [7] method is remarkable (see the top left plot of Figure 8). In this case, this method outperforms by more than one order of magnitude. However, its effectiveness decreases significantly 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 of a given collision (of a minimum of n bits) approaches for genuine and impostor comparisons. The approach of Gadde *et* al. [6] had the poorest performance for all the environments, whereas the method of Mukherjee and Ross [15] ranked third for the range of the performance space in good-quality environments. However, this was the unique technique that did not attain hit values above 0.9, either for good-quality or degraded data.

The analysed method ranked second on good-quality data and showed the least decrease in performance for degraded data. The higher robustness was particularly evident for very high hit rates, which is the most important range for biometrics scenarios. Table 2 summarizes the performance indicators and the corresponding 95% confidence intervals for the three types of environments. Each cell contains two values: the top value regards the full operating range, and the bottom values are

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

for the hit $\geq 95\%$ range. Again, the values confirm the above observations about the relative performance of the techniques analyzed.

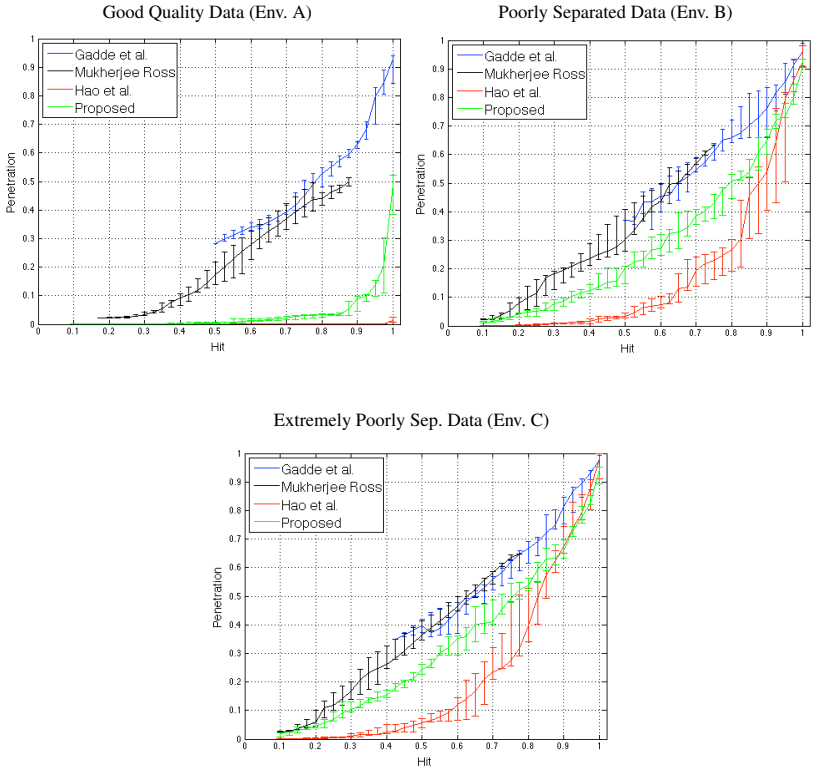

Fig. 8 Comparison between the hit / penetration rates observed for the strategy focused in this chapter and three state-of-the-art methods used as baselines. Results are expressed for three levels of data quality.

Figure 9 shows a statistic of the penetration rates (vertical axes) observed for queries that returned the true identity, for five kinds of environments, ranging from poorly separated (Env. C) to good quality data (Env. A). This plot emphasizes the good performance of the Gadde *et* al. method for good-quality data, obtaining penetration values close to 0. For poor-quality data, though the median value of the anlysed method's data penetration is higher than that of Hao *et* al. (≈ 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 for our proposal than for any other. Additionally, the inter-quartile range of Proenca's method penetration values was narrower than that of Hao *et* al.'s method, which is a signal of the stability of its performance with respect to different queries. For all methods tested, the penetration

Table 2 Summary of the performance indicators in synthetic data, with respect to four other strategies used as comparison terms. 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
Proença [17]	0.91 ± 0.01 0.90 ± 0.01	0.12 ± 0.01 0.15 ± 0.01	0.67 ± 0.02 0.50 ± 0.02	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. [7]	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. [6]	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 [15]	0.67 ± 0.01 -	0.48 ± 0.00 -	0.59 ± 0.03 -	0.58 ± 0.03 -	0.57 ± 0.01 -	0.60 ± 0.01 -

values decrease substantially for good separable data, though this is less evident for Mukherjee and Ross's proposal. This decrease is explained by the intrinsic properties of the clustering process involved here: clusters tend to have similar number of elements, and for every query, all identities inside a given cluster are returned. This prevents only a small set of identities from being returned even for highly separable data.

Figure 10 shows a zoom-in of the hit / penetration rates for three environments. Based on these, it is evident that the method analysed in this chapter consistently outperforms all the others for high hit values (above 0.9). Additionally, it is the unique that obtained full hit values with penetration smaller than one, meaning that was the unique that always retrieved the true identity and simultaneously reduced the search space. The minimum hit value above which the analysed method becomes the best appears to be a function of the data separability. This is confirmed by the rightmost plot, which relates the quality of data and the minimum hit value. For the worst kind of data (Env. C, quality=0.0), the analysed method outperforms any other for hit values above 0.88. As data separability increases, the minimum hit value varies roughly linearly, and for environments with a quality index higher than 0.2, the method of Hao *et* al. starts to be the best and should be used instead of ours.

4.2 Well Separated Near Infra-Red Data

The CASIA-Iris-Thousand[2] was used in performance evaluation, to represent reasonably well separated data. This data set contains 20 000 images from both eyes of 1 000 subjects, yielding the evaluation with 2 000 different classes (eyes).

The noise-free regions of the irises were segmented according to the method of He *et* al. [8] and an elliptical parameterization was chosen for both iris boundaries, according to the random elliptic Hough Transform algorithm. Next, the reasonability of the segmentation was manually adjusted, 110 images were discarded due to

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

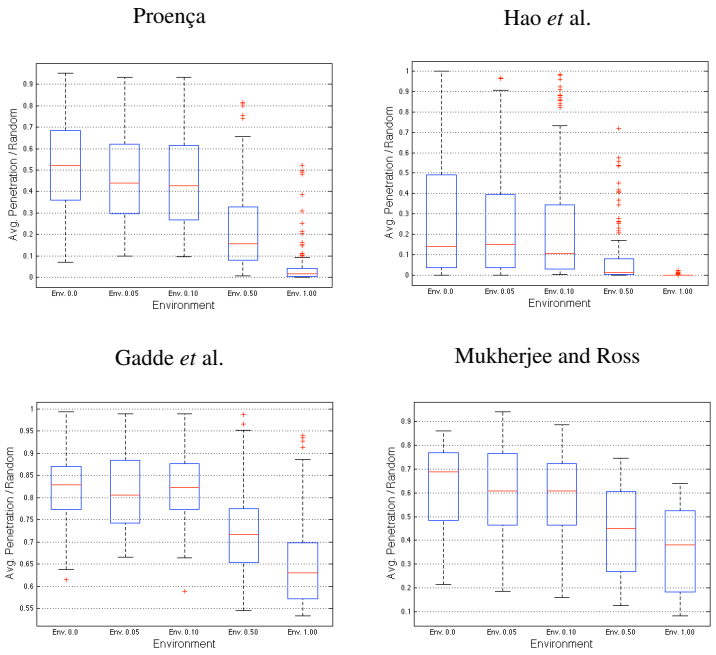

Fig. 9 Boxplots of the penetration rates observed in cases where the true identity was retrieved. Values are shown for different levels of data separability, starting from data of poorest quality (Env. 0.0) to good quality data (Env. 1.0).

bad quality and the remaining data translated into a pseudo-polar coordinate system using the Daugman's *rubber sheet* model. Next, three different configurations for Gabor kernels were used in signature encoding (wavelength ω and orientation θ were varied, phase ϕ and ratio r were not considered) . The optimal parameters for the Gabor kernels g 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.

$$g(x,y,\omega,\theta,\sigma_x,\sigma_y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}\left(\frac{\Phi_1^2}{\sigma_x^2}+\frac{\Phi_2^2}{\sigma_y^2}\right)} e^{i\frac{2\pi\Phi_1}{\omega}}, \tag{18}$$

being $\Phi_1 = x\cos(\theta) - y\sin(\theta)$, $\Phi_2 = -x\cos(\theta) + y\sin(\theta)$, ω the wavelength, θ the orientation and $\sigma_x = \sigma_y = \omega/2$. The parameters found were obtained by exhaustive evaluation in a training data set of 200 images randomly sampled from the data set: $(\omega,\theta) = \{(0.33,\pi/4),(0.28,3\pi/4),(0.51,\pi/2)\}$. Figure 11 gives some exam-

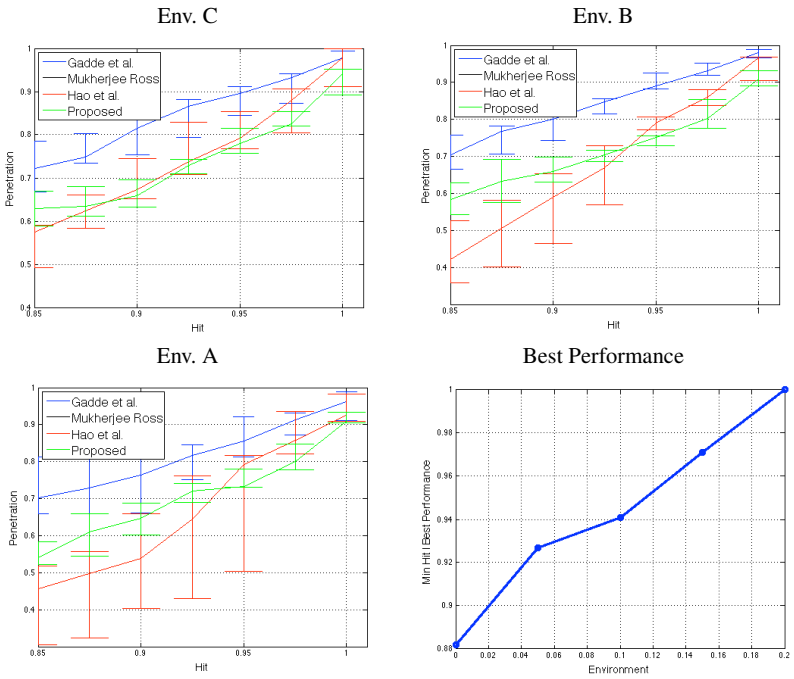

Fig. 10 Comparison between the hit/penetration plots in the performance range that was considered most important for biometric recognition purposes (hit values above 0.85). In poorly separable data the analysed method outperforms all the remaining ones in this performance interval, and the minimal hit value above which our method becomes the best varies roughly linearly with the data separability (bottom right plot).

ples of the noise-free iris masks and the iris boundaries for the CASIA.v4 Thousand images.

Results are given in Figure 12. The top left plot gives the decision environment, according to the recognition scheme used. At the center, a comparison between the hit / penetration values for the four techniques is shown, whereas the plot given at the top right corner summarizes the penetration rates in cases where the true identity was retrieved. Results confirm the previously obtained for synthetic data (for environments of average to good quality) and the approach of Hao *et* al. largely outperformed. The analysed method got a consistent second rank, followed by that of Mukherjee and Ross and Gadde *et* al's. The boxplots confirm these observations, being also notorious the smaller variance of the analysed method and Hao *et* al's in the number of retrieved identities, when compared to Gadde *et* al's and Mukherjee and Ross.

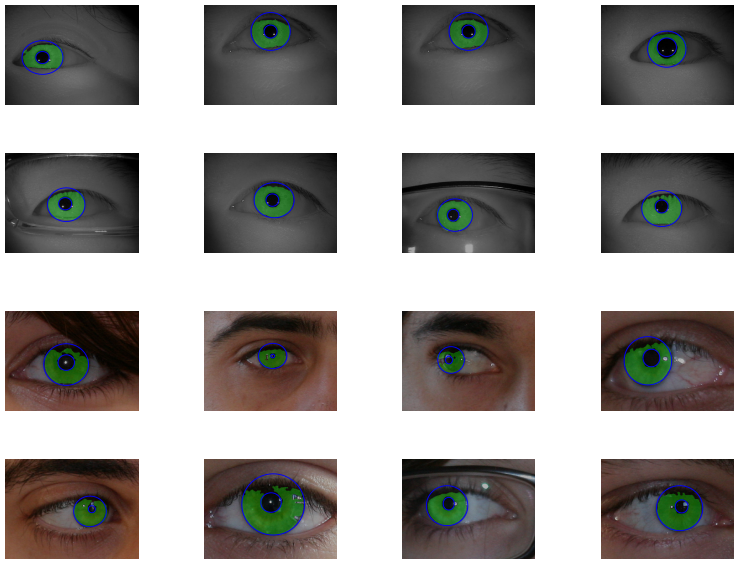

Fig. 11 Examples of the real iris images used in performance evaluation, segmented according to the method of He *et* al. [8]. The upper rows regard the CASIA.v4 Thousand data set, and the bottom rows give images of the UBIRIS.v2 data set.

4.3 Poorly Separated Visible Wavelength Data

The UBIRIS.v2 [16] data set constitutes the largest amount of iris data acquired from large distances (four to eight meters) at visible wavelengths, containing images of degraded quality that lead to poor separability between the matching scores of *genuine* and *impostors* comparisons. It has 11 102 images from 522 classes, from which 285 were not considered due to their extreme low quality level (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. [8] and followed the same processing chain, using the Gabor filters G with parameters $(\omega,\theta) = \{(0.18,\pi/6),(0.35,4\pi/6),(0.20,7\pi/8)\}$. The bottom rows of Figure 11 illustrate some examples of the images used.

These results were regarded in a particularly positive way, as they correspond to the environments where this method was devised for. As illustrated by the decision environment of Figure 13, classes have extremely poor separability, that can only be used in cases where no human effort is putted in the recognition process (e.g., automated surveillance). For this type of data, the analysed method outperformed in the most interesting performance range, i.e, for hit values above 90% (plot at the center). The rightmost plot gives a complementary perspective of results, by comparing the penetration rates in queries where the true identity was retrieved. In this case, the Proença's method got clearly higher penetration rates than Hao *et* al's, and

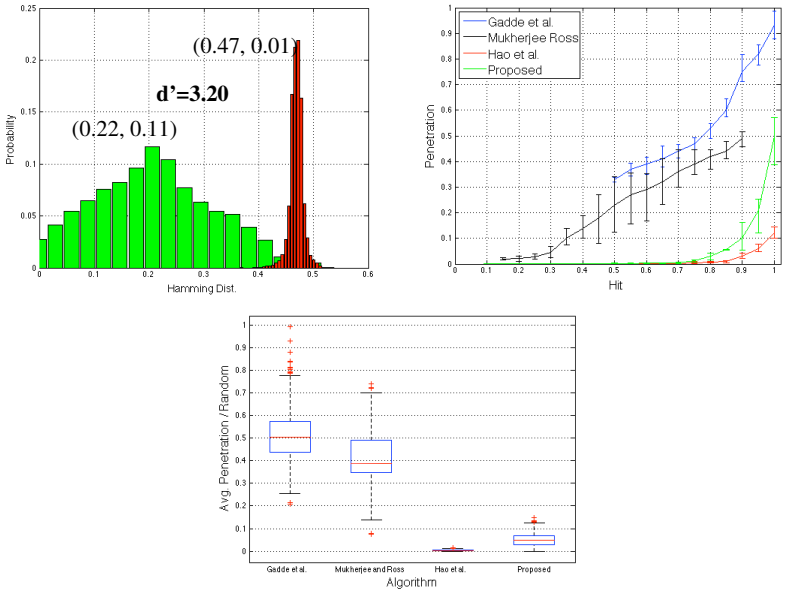

Fig. 12 Results observed for the CASIA.v4 Thousand iris data set. The top left plot gives an illustration of the decision environment yielded by the used recognition techniques is given. Plot at the top right corner compares the hit / penetration rates and the bottom plot summarizes the penetration rates observed in cases where the true identity was retrieved.

the value for the upper whisker is particularly important: for all queries the former method was able to reduce the set of identities retrieved, which did not happened in any of the other methods. Confirming the previous results, Hao *et* al.'s was the best for low hit values and got a solid second rate 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 the smaller variability with respect to different queries. Mukherjee and Ross' slightly better results than Gadde *et* al.'s, but in the former no hit values above 0.9 were obtained.

Table 3 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 meaningful range for biometrics scenarios (hit values above 95%). The values highlighted in bold confirm the suitability of the Proença's method to work on poorly separable data (UBIRIS.v2, $\Delta\gamma = +0.11$ of ours with respect to Hao *et* al.) and stress the effectiveness of Hao *et* al.'s method when working in scenarios that correspond to the currently deployed iris recognition systems (CASIA.v4 Thousand, $\Delta\gamma = -0.07$ of ours with respect to Hao *et* al.).

Acknowledgements This work was supported by PEst-OE/EEI/LA0008/2013 research program.

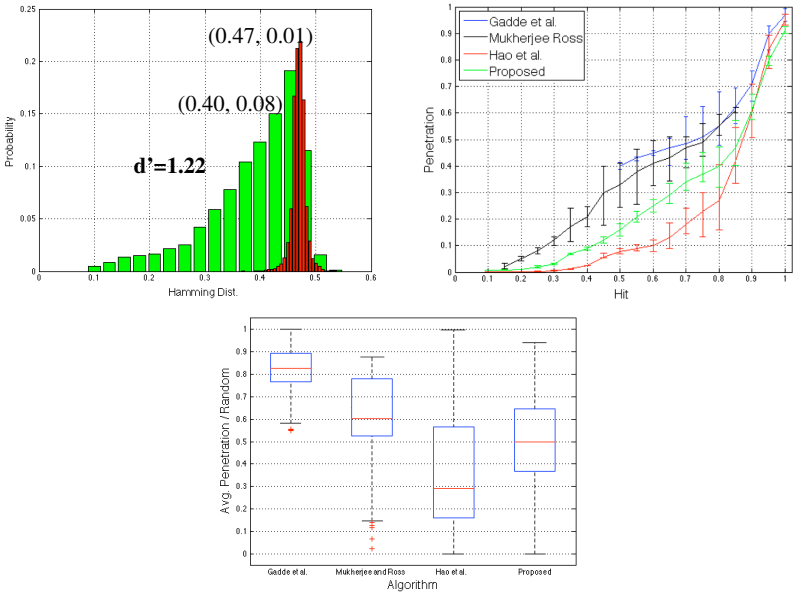

Fig. 13 Results observed for the UBIRIS.v2 iris data set. The top left plot givesan illustration of the decision environment yielded by the used recognition techniques is given. Plot at the top right corner compares the hit / penetration rates and the right plot summarizes the penetration rates observed in cases where the true identity was retrieved.

Table 3 Summary of the performance indicators (17) 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
Proença [17]	0.91 ± 0.02 $\mathbf{0.88 \pm 0.02}$	0.12 ± 0.01 $\mathbf{0.14 \pm 0.02}$	0.71 ± 0.02 $\mathbf{0.53 \pm 0.03}$	0.36 ± 0.02 $\mathbf{0.78 \pm 0.02}$
Hao *et* al. [7]	0.96 ± 0.01 $\mathbf{0.95 \pm 0.01}$	0.04 ± 0.01 $\mathbf{0.05 \pm 0.01}$	0.75 ± 0.03 $\mathbf{0.42 \pm 0.06}$	0.34 ± 0.02 $\mathbf{0.82 \pm 0.04}$
Gadde *et* al. [6]	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 [15]	0.76 ± 0.02 -	0.43 ± 0.02 -	0.61 ± 0.02 -	0.46 ± 0.02 -

5 Conclusions

This chapter aimed at summarising the state-of-the-art in terms of indexing / retrieving strategies for iris biometric data. In particular, we focused in the description of one technique to operate in *IrisCodes* extracted from low quality data, i.e., with a poor separability between the matching scores of genuine and impostor dis-

tributions. The described technique is based on the decomposition of the codes at different scales and in their placement in nodes of an n-ary tree. In the retrieval process, only portions of the tree are traversed before the stopping criterion is achieved. The main advantages of the described technique with respect to the state-of-the-art are three-fold: 1) the proposed technique has consistent advantages over other techniques when applied to poorly separated data, specifically in the performance range that is relevant for biometrics (hit values above 95%); 2) these levels of performance were achieved without a substantial increase in the computational burden, turning the use indexing / retrieving advantageous (in terms of turnaround time) when more than 54 000 identities are enrolled in the system; and 3) the method is quasi-independent of the iris signature encoding scheme, provided that it produces a binary signature.

References

1. H. Bay, A. Ess, T. Tuytelaars, L. Van Gool. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, vol. 110, issue 3, pag. 346-359, 2008.
2. 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.
3. J. Daugman and I. Mallas. Iris recognition border-crossing system in the UAE. *International Airport Review*, vol. 8, issue 2, pag. 49-53, 2004.
4. D. Donoho and I. Johnstone. Ideal Spatial Adaptation by Wavelet Shrinkage. *Biometrika*, vol. 81, pag. 425-455, 1994.
5. J. Fu and H. Caulfield and 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.
6. R. Gadde and 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.
7. F. Hao and 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.
8. Z. He and T. Tan and 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.
9. Institute of Automation, Chinese Academy of Sciences. CASIA Iris Image Database. `http://www.cbsr.ia.ac.cn/IrisDatabase`, 2009.
10. 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.
11. D. Lowe. Object recognition from local scale-invariant features . *Proceedings of the International Conference on Computer Vision*, pag. 1150-1157, 1999.
12. S. Mallat. A Wavelet Tour of Signal Processing. *Academic Press*, ISBN: 0-12-466606-X, 1999.
13. H. Mehrotra and 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.
14. H. Mehrotra and B. Srinivas and 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.
15. R. Mukherjee and A. Ross. Indexing Iris Images. *Proceedings of the 19th International Conference on Pattern Recognition (ICPR 2008)*, pag. 1-4, 2008.

16. 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.
17. H. Proença,. Indexing and retrieving Heavily Degraded Data. *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 12, pages 1975–1985, 2013.
18. 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.
19. X. Qiu and 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.
20. C. Rathgeb, F. Breitinger, H. Baier and C. Busch. Towards Bloom Filter-based Indexing of Iris Biometric Data. *Proceedings of the 8th IAPR International Conference on Biometrics (ICB 2015)*, pag. 422-429, 2015.
21. Unique Identification Authority of India. [online], `http://uidai.gov.in/about-uidai.html`, accessed on June, 2012.
22. Identity and Passport Service. [online], `http://www.direct.gov.uk/en/travelandtransport`, accessed on June, 2012.
23. M. Vatsa and 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.
24. L. Yu and 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.
25. 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.