Proc. of IEEE International Workshop on Information Forensics and Security (WIFS), (Seattle, USA), December 2010

Indexing Iris Images Using the Burrows-Wheeler Transform

Ravindra B. Gadde, Donald Adjeroh, Arun Ross

Lane Department of CSEE, West Virginia University
Morgantown, WV 26506, USA.
rgadde@mix.wvu.edu, don@csee.wvu.edu, Arun.Ross@mail.wvu.edu

***Abstract*—In most biometric identification systems, the input biometric data has to be compared against that of every identity in the database in order to determine the identity of the input. A major problem with this approach is the impact on response time which can increase significantly with the size of the database. In certain applications such as real time monitoring, this delay may not be acceptable. In this work, we propose a method for indexing iris images for rapid identity retrieval. Every entry in the database is assigned an index code based on which a small subset is retrieved and matched in response to a query. The basis of our approach is the sorted context property of the Burrows Wheeler Transform, a popular transformation used in data compression. Experiments on the CASIA version 3 iris database show a significant reduction in both search time and search space.**

***Index Terms*—Iris, Indexing, Burrows Wheeler Transform, Identification.**

I. Introduction

Biometric identification is the process of associating an identity to the input biometric data by comparing it against the enrolled identities in a database [1]. In many systems, this comparison is undertaken in an exhaustive manner, i.e., the input data of an individual is compared against *all* enrolled data in order to determine the identity of the individual. The false positive identification rate as well as the matching time can increase with the size of database [2]. So, a procedure that reduces the search space is necessary in order to increase the efficiency of the system.

In this work, we concern ourselves with the problem of iris recognition. Due to its rich texture and the observed statistical independence of the iris codes of different eyes (the iris code is the feature template extracted from the iris), the iris is considered to be a very reliable biometric modality for identification [3]. Thus, iris identification has been shown to be very effective in large databases. However, the processing and matching of non-ideal irides can be computationally expensive and, therefore, methods to speed up the identification process are necessary [4]. The search space of an identification operation can be reduced by rapidly choosing a subset of iris images from the database before matching. This can be accomplished using two methods: classification and indexing. In a classification scheme, the database is partitioned into a certain number of classes based on the physical and structural properties of the iris. During identification, the input image is assigned to one of the classes and is subsequently compared against those irides in the database which are in the same class. Existing classification schemes are mainly based on color [5] or texture [6], [7], [8] details of the iris. The main limitation of the classification approach is the potentially uneven distribution of the identities across the classes and the effect of noise in determining the correct class.

In the indexing approach, each iris is assigned with an index value. However, the index values of two iris images of the same individual may not be the same because the process of data acquisition and processing is subject to noise. Therefore, indexing systems retrieve those identities whose indices are similar to the index value of the input data. The input image is matched only against the retrieved identities thereby reducing the identification time and, potentially, the identification error rate. Indexing of iris images is often based on iris codes [9], [10] or key features such as SIFT [11]. Hao et al. [9] proposed a fast search algorithm for a large fuzzy database that stores iris codes. Mehrotra et al. [11] used SIFT key points to obtain indices of the hash table during indexing and retrieved a list of iris images from the hash table by casting votes. Work on indexing of iris images by Mukherjee et al. [10] used both iris codes and features extracted from the iris texture and demonstrated that about 80% of the input images could be correctly identified with a low penetration rate of 8%.

In this work, we present a new method for indexing iris images based on the context clustering property of the Burrows-Wheeler Transform (BWT). The next section provides a brief description of the BWT. Section III then describes the general procedure for indexing iris images. Results are presented in IV. Section V concludes the paper.

II. The BWT and Iris Indexing

The Burrows-Wheeler Transform (BWT) [12] performs a permutation of the characters in a sequence, such that characters in lexically similar contexts are in proximity. Given an input text string $T = t_1, t_2....t_m$ over some finite alphabet $\Sigma = \alpha_1, \alpha_2....\alpha_{|\Sigma|}$, the forward BWT is done in three steps:

1) Form m permutations of T by cyclic rotations of the characters in T. The permutations form a $m \times u$ matrix $M^{'}$, with each row in $M^{'}$ representing one permutation of T;

WIFS'2010, December 12-15, 2010, Seattle, WA, USA.
978-1-4244-9080-6/10/$26.00

2) Sort the rows of $M^{'}$ lexicographically to form another matrix M. M includes T as one of its rows;
3) Record L, the last column of the sorted permutation matrix M, and id, the row number for the row in M that corresponds to the original sequence T.

As an example, suppose T=**mississippi**. Let F and L denote the array of first and last characters in M respectively (see Table. I). Then, F=**iiiimppssss** and L=**pssmipissii**. The output of the transformation will be the pair: (L, id)=(**pssmipissii**, 5). Generally, the effect is that the contexts that are similar in the input sequence T are brought closer together in the output sequence L. (Notice how all the rows that started with the same context, i.e., same starting symbols, are grouped together in L, e.g. the rows that start with **issi, ssi,** etc.). This similarity in nearby contexts is the key to improved compression performance in BWT-based compression systems [12], [13]. The BWT is reversible. It is quite striking that given only the (L, id) pair, the original sequence can be recovered exactly. Adjeroh et al. presented a detailed treatment of the BWT in the recent book [13], including its performance, connection with suffix trees and suffix arrays, and various applications. In this work, our focus is on studying the potential of the sorted contexts of the BWT in indexing iris images.

TABLE I
BWT FORWARD TRANSFORMATION FOR $T = mississippi$. $M^{'}$ IS THE MATRIX OF CYCLIC ROTATIONS. M IS THE MATRIX AFTER SORTING. * SHOWS THE ROW IN M CORRESPONDING TO THE ORIGINAL SEQUENCE.

$M^{'}$	M	(F)	(L)
mississippi	imississipp	i	p
ississippim	ippimississ	i	s
ssissippimi	issippimiss	i	s
sissippimis	ississippim	i	m
issippimiss	mississippi	m	i*
ssippimissi	pimississip	p	p
sippimissis	ppimississi	p	i
ippimississ	sippimissis	s	s
ppimississi	sissippimis	s	s
pimississip	ssippimissi	s	i
imississipp	ssissippimi	s	i

A. BWT Context Partitions

With the suffix sorting stage of the BWT, suffixes that are similar in the original sequence will be placed together in the sorted matrix of rotations of the BWT. Thus the BWT output symbols in the same region of the output array L are likely to have similar following suffixes, that is, similar forward contexts in the original sequence. Therefore, the output stream can be partitioned into different segments based on the similarity in the symbol contexts. For example, consider the sequence T=mississippi used in Table I, with L=pssmipissii. For order-1 context partitions, all the symbols in the L array with the same starting symbol in the corresponding rows in the F array will be in the same context. Thus, the first four symbols (p,s,s,m) in L will be in the same context partition - the i-partition. We can exploit this context clustering (or partitioning) property of the BWT in indexing iris images.

Consider the iris image after preprocessing to remove the effect of, say, illumination and noise in the image. Suppose we can locate patches in the iris image that are similar in their textural characteristics and that occur a certain number of times in the iris [10], [14]. We speculate that the geometric relation between the positions of occurrence of the patches in the iris image along with the specific pattern within these patches can form the basis for constructing an effective index for the iris images. The key challenge is to develop a fast method for determining such repeated patches and for locating the positions of occurrence of such patches on the iris image. This can be viewed as a pattern matching problem, which is a major application of the BWT [13]. In this work, we use the BWT as the basis for identifying the repeated patches and their positions of occurrence in the iris image.

III. Proposed Approach

The iris exhibits a very rich texture due to the numerous anatomical entities on its stroma. The texture in an iris is believed to be relatively stable (a claim that has been challenged in the recent literature) and have been observed to vary in its intricate detail from one eye to another [15]. This facilitates the design of an identification system based on this texture. Apart from recognition, the iris texture can also be used for indexing. In this section, we design an indexing scheme based on the distribution of the patterns formed because of the texture.

A classical iris recognition system has an image acquisition module, an iris localization or segmentation module, a normalization module, an encoding module and a matching module. The acquisition module obtains an image of the iris in the near-infrared (NIR) spectrum. During segmentation, the iris is localized and isolated from the sclera, pupil, eyelids and eyelashes. In the normalization module, the segmented iris is unwrapped and converted from the cartesian domain into a doubly dimensionless (polar-like) domain resulting in a rectangular entity that is subjected to subsequent processing. Normalization is performed using Daugman's rubber sheet model [15]. Fig. 1(a) shows an example of a normalized iris image.

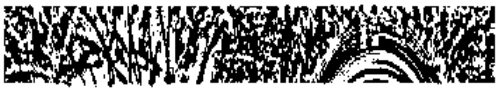
In the proposed algorithm, the normalized gray scale iris image is first converted to a binary image. Next, a certain horizontal n-bit pattern is chosen and the locations of these patterns are found in the iris image. Then, the normalized iris image is divided into vertical segments and, based on the maximum occurrence of the n-bit pattern among these segments, the iris image is assigned an index value based on the segment number.

A. Pre-Processing

In cases where thresholded images are used, changes in illumination can have a significant impact on the results. Thus, a pre-processing technique is used to mitigate this problem.

(a)

(b)



(c)

Fig. 1. (a) Example of a normalized iris image. (b) Normalized iris image after filtering. (c) Normalized binary iris image

1) Illumination variation: The iris patterns are relatively sensitive to illumination variations, especially when dealing with binary images. To overcome this limitation, we adapt the color ratio model [16] to generate illumination-invariant versions of the iris images.

Assume that the illumination variation in a scene is such that the illumination is locally constant for a given spatial position in the image. Then, the average intensity value over a small region should not change much within the region. Thus, the ratio of a pixel value in a region to the mean intensity value over the region or window (for instance, using a small window of size W) should be close to one.

Let $h(x, y)$ be a window of size WxW around the pixel $I(x, y)$. The color ratio[1] is defined as follows:

$$R(x, y) = \frac{I(x, y)}{\mu(h(x, y))}, \tag{1}$$

where $\mu(h(x, y))$ is the mean pixel intensity in the window.

However, when there is a definite boundary (or edge variations), the ratio will move away from 1. Thus, essentially, the ratio encodes the boundary variations (or edge variations) in the image. Because such boundaries are invariant to illumination changes, we expect a robust result against potential illumination variations in the iris image. Fig. 1(b) shows the normalized iris image of Fig. 1(a) after filtering using the color ratio model.

2) Conversion to binary image: To convert the gray scale image to binary image the cumulative distribution function of the normalized iris image is calculated. Based on the cumulative distribution function, a threshold is determined and values of pixels greater than this threshold are considered as '1' and those less than this threshold are considered as '0'.

Consider a discrete gray scale image $J(x)$ and let t_i be the number of occurrences of gray level i. The probability of occurrence of a pixel of level i in the image is given by

$$p_x(i) = p(J(x) = i) = \frac{t_i}{t} \qquad 0 \leq \ i < L, \tag{2}$$

where L is the total number of gray levels in the image and t is the total number of pixels in the image. Here, $p_x(i)$ represents the normalized histogram of pixel value i.

The cumulative distribution function corresponding to p_x is given by

$$F_x(i) = \sum_{j=0}^{i} p_x(j). \tag{3}$$

Now consider, the value of i where $F_x(i)$ is $\frac{1}{2}$. The values of pixels above i are considered as '0' and below i are considered as '1'. The result of this procedure on the example iris image of Fig. 1(b) is shown in Fig. 1(c).

B. Pattern Finding

After obtaining the binary image, we identify a horizontal n-bit binary pattern which is consistent among all the images. This can be done by choosing the pattern with the least value of coefficient of variation. The coefficient of variation (CV) for a particular n-bit pattern is simply the ratio of the mean and standard deviation of the number of occurrences of that pattern across all the training images. The coefficient of variation for each of the 2^n patterns has to be calculated using the same training images and then a pattern (say P_i) with the least CV has to be chosen. Thus, the key problem is to count the number of occurrence of each n-bit horizontal pattern in the normalized binary image and to locate the positions of each occurrence. This is essentially a problem of pattern matching, which is performed using the BWT.

First, the 2D binary image is converted into a 1D sequence by simply concatenating the rows, from the first row to the last. Each row is terminated using a special character, such that the search for patterns will not report patterns that span two different rows. Then, the 1D sequences from the individual iris images in the database are further concatenated to form one long sequence, T. A special end of file marker is used to indicate the end of each iris image. The BWT is then applied to T, the database sequence, producing the L array, and an index id as the output. Pattern matching is then performed on the database via the BWT output pair (L, id). Notice that since we know the size of each iris image, and the rows are of fixed length, a position in T uniquely determines the corresponding image and the (x, y) position in that image.

[1]Although the term "color" is used for historical purposes, the measure in this paper is used on grayscale images

Algorithm 1. String comparison function for Binary Search

BINARY-SEARCH-STRCMP(P, W, L, i)
1 $m \leftarrow$ LENGTH(P); $j \leftarrow 1$; $i \leftarrow W[i]$;
2 **while** $m > 0$ **and** $L[i] = P[j]$ **do**
3 $\quad i \leftarrow W[i]$; $m \leftarrow m - 1$; $j \leftarrow j + 1$;
4 **end while**
5 **if** $m = 0$ **then** **return** 0;
6 **else** **return** $P[j] - L[i]$;
7 **end if**

Consider an n-bit binary pattern, P. To search for P in T (via L), we use an auxiliary mapping array, W defined as follows:

$$\forall i : 1 \leq i \leq n, T[i] = L[W^i[id]]$$

where $W^1[x] = x$, $W^{i+1}[x] = W[W^i[x]]$, and id is the position in F of the first character of the text.

W provides a one-to-one mapping between the BWT F and L arrays, viz., $F[i] = L[W[i]]$. Using the earlier example with mississippi, we will have W = [5, 7, 10, 11, 4, 1, 6, 2, 3, 8, 9]. Construction of W is performed in linear time with respect to the length of T.

The sorted list of suffixes of T are available from M, the sorted matrix of rotations. If a pattern appears in the database T, it must appear at the start of one or more rows in M. Given that the rows in M are sorted, all occurrences of such a pattern will be located next to each other in M. In practice, we access M via F, the array of first characters. F in turn is accessed from L (the BWT output) via the mapping array W. Then, the search is performed using a modified binary search algorithm [17], [13], using L, W, and id. The procedure below shows the comparison function used by the modified binary search algorithm, where i denotes the row in the rotation matrix M being compared to the pattern. Let s be the string representation of the row. The function will return 0 if P is a prefix of s, < 0 if $P < s$, and > 0 if $P > s$

Using the above, we search for each valid n-bit pattern in T (the database), and report the occurrence counts and the locations in T where each occurred. We then determine the patterns with least coefficient of variation in their occurrence counts over all the images in the database. In this work, we selected two patterns with the least values for CV, denoted P_1 and P_2, as the basis for our index. Fig. 2 shows the positions of occurences of pattern P_1 in the sample binary image used in the running example.

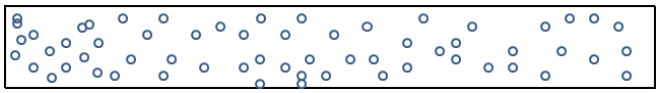


Fig. 2. Positions of pattern P_1 on the normalized binary image.

C. Indexing

The indexing scheme has two stages: i) Training and ii) Testing.

i) Training is an offline process, in which every training image, x, in the database is assigned a particular index I_x.

ii) Testing is an online process, in which the probe image, y, is assigned a particular index I_y. The probe image is then matched only with those set of images in the database with I_y as their index value, along with those with indices similar to I_y.

In this work, the indexing scheme is based on the *count* of occurrence of the n-bit binary patterns (say P_1) in the normalized binary iris image (x). Thus, during training, all the positions where P_1 occurred in image x are identified. The image is divided into K segments along the vertical (i.e., radial) direction. The iris image is then assigned an index with segment number I_x, where the segment I_x has the maximum number of occurrences for pattern P_1. The division of the iris image into segments in the radial direction compensates for the affine transformations due to changes in viewing angle. During testing, a procedure similar to that used for training is followed. However, in this case instead of assigning a single index corresponding to a single segment number I_x, the image is assigned four indices which correspond to the top 4 segments with maximum counts of occurrence of pattern P_1. Now, the test image (probe) is compared only against those images in the database that have these 4 indices as their index, starting with the first index. Fig. 3 shows the basic indexing scheme described above.

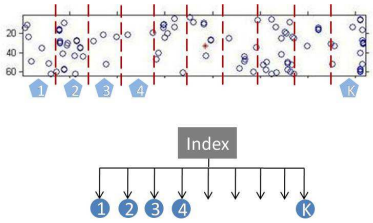


Fig. 3. The proposed indexing scheme. The vertical regions correspond to the segments numbered 1 to K.

IV. RESULTS

The CASIA version 3 iris image database was used in the experimental evaluation of the iris indexing technique proposed here. The images in this database were segmented and normalized using the algorithms proposed by Shah and Ross [18]. In this work, the left normalized iris images are used which contain images of resolution 360 x 64 pixels pertaining to 249 subjects. In some cases, the normalized images of a subject were not successfully generated by the algorithm. Only those users with at least two normalized sample images per eye were considered in this work (a total of 189 users). The images for each subject were equally divided into training and test sets if the number of images for that subject was even. The training set had an extra image from a subject if the number of images for that subject was odd.

Experiments are conducted using 4-bit and 8-bit binary patterns. The two patterns P_1, P_2 with the least values of coefficient of variation are chosen (P_1 has the overall least value) as per our discussion in section III-B. We studied the performance of various window sizes in the color-ratio model, viz., W=3, 5, 7 and for different number of segments, viz., K= 4, 6, 8, 10,1 2. The hit rate (R_h) and penetration rate (R_p) for different combinations of W and K were noted to study the performance of the technique. In order to find the optimal trade-off between hit rate(R_h) and penetration rate (R_p), we defined a new variable γ, which combines both the hit rate and penetration rate:

$$\gamma = \sqrt{(R_h) * (1 - R_p)}. \quad (4)$$

Experiment 1: The performance of the indexing scheme for 4-bit patterns P_1 and P_2 can be seen in tables II and III, respectively. For pattern P_1, the best value of γ = 90.90% occurred at a hit rate of 99.83% and a penetration rate of 17.23% with W=3 and K=10. For pattern P_2, the best value of γ = 84.52% occurred at a hit rate of 99.50% and a penetration rate of 28.19% with W=3 and K=8.

TABLE II
PERFORMANCE OF THE PROPOSED INDEXING SCHEME FOR A 4-BIT PATTERN (P_1)

	W=3		W=5		W=7	
	R_h	R_p	R_h	R_p	R_h	R_p
K=4	100	23.51	100	26.24	100	48.96
K=6	100	21.77	100	36.44	100	57.82
K=8	99.83	21.83	100	39.18	100	59.56
K=10	**99.83**	**17.23**	100	26.80	100	33.62
K=12	99.17	18.61	100.0	28.52	100.0	42.16

TABLE III
PERFORMANCE OF THE PROPOSED INDEXING SCHEME FOR A 4-BIT PATTERN (P_2)

	W=3		W=5		W=7	
	R_h	R_p	R_h	R_p	R_h	R_p
K=4	100	29.38	100	39.58	100	48.44
K=6	**99.50**	**28.19**	100	48.39	100	51.50
K=8	100	32.30	100	55.26	100	55.84
K=10	99.66	39.55	100	56.84	100	59.26
K=12	99.83	38.82	100.0	58.44	100.0	59.31

Experiment 2: Similar experiments were conducted for 8-bit patterns with varying W and K. The performance of the indexing scheme with varying window sizes W and for different values of K for patterns P_1 and P_2 can be seen in tables IV and V, respectively. For pattern P_1, the best value of γ = 89.30% occurred at a a hit rate of 98.5% and a penetration rate of 19.03% with W=3 and K=10. For pattern P_2, the best value of γ = 89.09% occurred at a hit rate of 96.68% and a penetration rate of 17.90% with W=3 and K=10.

Experiment 3: In order to test the robustness of the algorithm, noise was added to the binary images by randomly inverting 50% of the bits for the images in the test set and repeating the aforementioned experiments. The performance of the indexing scheme with varying window sizes W and number of segments K for patterns P_1 and P_2 are shown in tables VI,VII,VIII and IX.

TABLE IV
PERFORMANCE OF THE PROPOSED INDEXING SCHEME FOR AN 8-BIT PATTERN (P_1)

	W=3		W=5		W=7	
	R_h	R_p	R_h	R_p	R_h	R_p
K=4	100	23.26	100	23.96	100	25.43
K=6	99.66	20.99	99.50	24.77	99.83	25.65
K=8	99	19.54	99.33	23.34	97.68	24.05
K=10	**98.50**	**19.03**	96.19	21.88	95.19	22.89
K=12	96.52	19.66	95.69	20.04	93.54	23.62

TABLE V
PERFORMANCE OF THE PROPOSED INDEXING SCHEME FOR AN 8-BIT PATTERN (P_2)

	W=3		W=5		W=7	
	R_h	R_p	R_h	R_p	R_h	R_p
K=4	100	24.28	100	57.73	100	59.65
K=6	99	22.49	100	47.16	100	55.95
K=8	98.01	21.17	99.83	40	100	48.54
K=10	**96.68**	**17.90**	100	38.79	100	45.28
K=12	100	18.52	100	31.36	100	33.84

TABLE VI
NOISY TEST DATA: PERFORMANCE OF THE PROPOSED INDEXING SCHEME FOR A 4-BIT PATTERN (P_1)

	W=3		W=5		W=7	
	R_h	R_p	R_h	R_p	R_h	R_p
K=4	100	32.01	100	43.08	100	42.56
K=6	**99**	**27.13**	88.24	38.19	85.76	41.40
K=8	97.51	29.20	75.99	36.32	77.64	34.03
K=10	89.23	26.66	65.39	30.41	69.70	29.71
K=12	88.74	25.29	61.09	27.87	58.60	29.13

TABLE VII
NOISY TEST DATA: PERFORMANCE OF THE PROPOSED INDEXING SCHEME FOR A 4-BIT PATTERN (P_2)

	W=3		W=5		W=7	
	R_h	R_p	R_h	R_p	R_h	R_p
K=4	100	34.99	100	44.24	100	55.23
K=6	**92.54**	**28.51**	99	48.36	99..66	55.33
K=8	77.15	23.51	96.19	51.92	99.50	55.25
K=10	70.19	19.37	97.68	54.4	99.17	56.36
K=12	61.09	16.48	97.68	56.45	99.17	56.54

From the above experiments, we observe that the hit rates and penetration rates using 4-bit and 8-bit patterns are affected when the binary images are very noisy. Currently, we are looking at ways to impart noise resilience to the index patterns.

V. CONCLUSION

In this paper we address the problem of iris indexing. The proposed algorithm is a fast searching technique that uses Burrows-Wheeler Transform to perform iris indexing. Compared to the previous work by Mukherjee et al. [10] (with hit rate = 80% and penetration rate = 8%, with γ=85.79%), our method showed significant improvement in performance

TABLE VIII
NOISY TEST DATA: PERFORMANCE OF THE PROPOSED INDEXING SCHEME FOR AN 8-BIT PATTERN (P_1)

	W=3		W=5		W=7	
	R_h	R_p	R_h	R_p	R_h	R_p
K=4	100	27.59	100	23.28	100	24.93
K=6	98.34	24.16	99.33	22.38	100	22.62
K=8	96.19	21.28	98.67	21.20	98.67	19.40
K=10	92.21	20.70	**96.68**	**19.09**	97.35	17.35
K=12	85.43	19.80	93.70	17.11	96.02	16.81

TABLE IX
NOISY TEST DATA: PERFORMANCE OF THE PROPOSED INDEXING SCHEME FOR AN 8-BIT PATTERN (P_2)

	W=3		W=5		W=7	
	R_h	R_p	R_h	R_p	R_h	R_p
K=4	100	27.56	100	35.02	100	41.36
K=6	98.50	20.12	93.70	30.25	90.23	35.26
K=8	96.68	21.13	85.59	28.24	79.47	32.29
K=10	**95.19**	**18.42**	84.27	23.78	71.35	27.81
K=12	93.04	16.26	79.13	22.63	70.69	24.23

with a 99.83% hit rate at a 17.23% penetration rate, with γ=90.90%. Further work is required to reduce the penetration rate, for instance, by using an alternate searching mechanism which combines different index patterns (P_1, P_2) of the same length. We are also exploring the use of iris codes in the BWT framework discussed in this paper. Future work would involve designing appropriate data structures to organize the indices generated for the iris images. This can potentially lead to an efficient retrieval strategy for very large databases.

ACKNOWLEDGMENT

This work was supported by US NSF CAREER grant number IIS 0642554.

REFERENCES

[1] A. Jain, A. Ross, and S. Prabhakar, "An Introduction to Biometric Recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 4–20, January 2004.

[2] A. Mhatre, S. Palla, S. Chikkerur, and V. Govindaraju, "Efficient search and retrieval in biometric databases," in *SPIE Defence and Security Symposium*, 2005, vol. 5779, pp. 265–273.

[3] J. Daugman, "High confidence visual recognition of persons by a test ofstatistical independence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1148–1161, 1993.

[4] A. Ross, "Iris Recognition: The Path Forward," *IEEE Computer*, vol. 43, no. 2, pp. 30–35, 2010.

[5] X. Qiu, Z. Sun, and T. Tan, "Global texture analysis of iris images for ethnic classification," *Advances in Biometrics*, vol. 3832, pp. 411–418, 2005.

[6] L. Yu, D. Zhang, K. Wang, and W. Yang, "Coarse iris classification using box-counting to estimate fractal dimensions," *Pattern Recognition*, vol. 38, no. 11, pp. 1791–1798, 2005.

[7] X. Qiu, Z. Sun, and T. Tan, "Coarse iris classification by learned visual dictionary," *Advances in Biometrics*, vol. 4642/2007, pp. 770–779, 2007.

[8] A. Ross and M. SamSunder, "Block Based Texture Analysis for Iris Classification and Matching," in *IEEE Computer Society Workshop on Biometrics at the Computer Vision and Pattern Recogniton (CVPR) Conference*, June 2010.

[9] 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, pp. 203–212, 2008.

[10] R. Mukherjee and A. Ross, "Indexing iris images," in *19th International Conference on Pattern Recognition*, 2008, pp. 1–4.

[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, no. 3, pp. 300–313, May 2010.

[12] M. Burrows and D.J. Wheeler, *A block-sorting lossless data compression algorithm*, Technical Report: Digital Equipment Corporation, Palo Alto, CA, 1994.

[13] D. Adjeroh, T. Bell, and A. Mukherjee, *The Burrows-Wheeler Transform: Data Compression, Suffix Arrays, and Pattern Matching*, Springer-Verlag New York Inc, 2008.

[14] S. Makthal and A. Ross, "Synthesis of Iris Images using Markov Random Fields," in *Proc. of 13th European Signal Processing Conference (EUSIPCO)*, 2005.

[15] J. Daugman, "How iris recognition works," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, 2004.

[16] D. A. Adjeroh and M. C. Lee, "On ratio-based color indexing," *IEEE Transactions on Image Processing*, vol. 10, no. 1, pp. 36–48, 2001.

[17] A. Firth, T. Bell, A. Mukherjee, and D. Adjeroh, "A comparison of BWT approaches to string pattern matching," *Software: Practice and Experience*, vol. 35, no. 13, pp. 1217–1258, 2005.

[18] S. Shah and A. Ross, "Iris Segmentation Using Geodesic Active Contours," *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 4, pp. 824–836, December 2009.