An Iris Retrieval Technique Based on Color and Texture

Umarani Jayaraman*
Indian Institute of Technology
Dept of Computer Sc. & Engg.
IIT Kanpur, Kanpur
umarani@cse.iitk.ac.in

Surya Prakash
Indian Institute of Technology
Dept of Computer Sc. & Engg.
IIT Kanpur, Kanpur
psurya@cse.iitk.ac.in

Phalguni Gupta
Indian Institute of Technology
Dept of Computer Sc. & Engg.
IIT Kanpur, Kanpur
pg@cse.iitk.ac.in

ABSTRACT

This paper proposes an efficient indexing scheme that can be used for retrieval from a large iris database. For a given color iris query image, the proposed indexing scheme makes use of iris color to determine an index and uses this index to reduce the search space in the large iris database. Further, for query q, the retrieval technique uses iris texture to find the top best match from the reduced search space. The proposed technique has been tested on two publicly available color iris databases, viz UPOL [10] of 384 images and UBIRIS [13] of 1860 fully noisy images and is found to be robust against change in gaze, illumination, partial occlusions and scale. In both the databases, the test reveals that a small subspace is sufficient to achieve 100% *hitrate* for the top best match under various scales, illumination and partial occlusion. The performance of the proposed indexing scheme is analyzed against the group based color indexing scheme proposed in [14]. The results show that proposed indexing scheme is performing better as compared to group based color indexing scheme with respect to *hitrate*, *penetration rate* and *CMC* curve.

Keywords

Iris Color Indexing, Iris Texture Retrieval

1. INTRODUCTION

The iris based biometric system provides an automated method to authenticate an individual with the help of iris textural patterns. For a query iris image, the problem of iris identification system is to find the top t best matches in the database consisting of N iris images. For a large database, it needs time to make $1 : N$ searches. In order to design a system which is more powerful and fast, the matching engine need to search in a reduced space in the database.

In the literature there exists very few work on indexing of iris database. Yu et al. [17] have proposed a coarse iris classification technique using fractals that classifies iris images into four categories. This classification technique reduces the computation time but compromises with the identification accuracy. Mukherjee and Ross [11] have proposed an indexing scheme for iris database in which block-based statistics is used. This indexing scheme is tested on CASIA version 3.0 iris database. Vatsa et al. [16] have proposed an indexing scheme which follows two steps process. The technique first uses Euler code to generate a small subset of possible matches. In the second step, it uses 2ν-SVM match score fusion algorithm to find the best matches from the list of possible matches obtained in the first step. However, all the above proposed indexing scheme works on gray scale iris images.

The indexing scheme proposed in this paper is for color iris images where indexing is based on iris color. The first attempt to index iris images using color has been made by Fu et al. [7]. This technique is based on artificial color and has used a set of nine artificial color filters to narrow down the search space. However, the artificial color filtering is not an effective approach to index because the colors are generated artificially which are very much different from the natural iris color. Further, the performance of the indexing scheme has not been tested on any publicly available databases. Another attempt to index iris images using color has been made by Puhan and Sudha [14] where they have proposed group based color indexing scheme which relies on the natural iris color. This technique first converts the iris color images from RGB space to YC_bC_r color space and computes two types of color indices, namely blue and the red indices using C_b and C_r components respectively. The range of values of red color indices and blue color indices are partitioned individually into some bins (groups). Depending upon the value of red and blue color indices, an image is assigned to one of these groups. During searching, for a query image a few groups from blue and red color indices are selected based on the blue and red color indices of query image. The nearest identities are declared based on the intersection between these groups. However, the problem with this group based indexing scheme is that a minor variation in the red and blue color values may lead an image being assigned to a different group. Further, this group based indexing scheme is solely dependent on color indices for recognising the iris images which may not be always helpful to distinguish the iris images of different subjects.

This paper proposes an indexing scheme which uses iris color for indexing and a retrieval technique based on iris texture to improve the performance of iris recognition sys-

*Corresponding author

ICVGIP '10, December 12-15, 2010, Chennai, India

tem. It uses Kd-tree [3] to index the color indices and SURF [1] features to retrieve the correct identity in the top best match. The paper is organised as follows. Preliminaries is given in Section 2. The proposed iris indexing scheme has been discussed in Section 3. Through this indexing scheme, the search space of the large iris database has been reduced effectively. In order to obtain the correct identity in the top best match from the reduced search space, an iris retrieval technique which uses iris texture has been proposed in Section 4. Performance of the proposed indexing scheme has been analysed against the group based color indexing scheme [14] in the next section. Conclusion is given in the last section.

2. PRELIMINARIES

This section discusses the preliminaries which are required in developing the proposed indexing technique. Section 2.1 describes the Kd-tree data structure which is used to index the color indices of iris by forming Kd-tree. Section 2.2 describes SURF feature extraction algorithm that is used to extract texture features from iris.

2.1 Kd-tree

The proposed indexing technique is based on the Kd-tree data structure [3, 15, 4]. This section discusses the salient features of Kd-tree. It is a binary tree that represents a hierarchical subdivision of space using splitting planes that are orthogonal to the coordinates axes. Kd-tree is a space-partitioning data structure for organizing points in a k dimensional space. Any application in which features are generated as multi-dimensional is a potential application for Kd-tree.

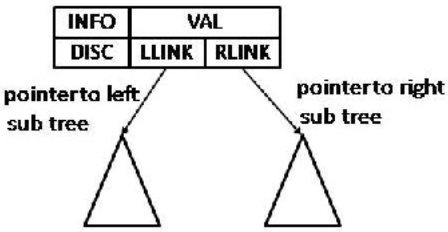


Figure 1: Structure of Kd-tree node

Structure of a node in Kd-tree is given in Fig. 1. Each node in Kd-tree consists of five fields. Node contains two pointers known as $LLINK$ and $RLINK$, which pointing to left subtree and right subtree respectively, if exists. Otherwise, it points to $null$. The field VAL is an array of length k containing real feature vector. The $INFO$ field contains descriptive information about the node. The $DISC$ field is a discriminator, which is an integer between 1 and k, both inclusive. In general, for any node P in the Kd-tree, let i be $DISC(P)$ and is defined as $level(P) \mod k$. Then for any node L in $LLINK(P)$, $L.VAL[i] < P.VAL[i]$; likewise, for any node R in $RLINK(P)$, $R.VAL[i] \geq P.VAL[i]$. All nodes on any given levels of the tree have the same discriminator. The root node has discriminator 1, and its two sons have discriminator 2, and so on to the k^{th} level on which the discriminator is k. Again the $(k+1)^{th}$ level has discriminator 1, and the cycle repeats; In general, next discriminator denoted as $NEXTDISC$, is a function defined as

$$NEXTDISC(i) = (i+1) \mod k$$

Number of nodes in the Kd-tree are same as the number of templates in the input file to be inserted in the tree. As it is mentioned already that k is the dimensionality of the template.

In order to insert a node P having the data into the Kd-tree, it starts searching from the root of the Kd-tree and finds its appropriate position where the node can be inserted. Bentley [3] shows that the average cost of inserting and searching a node in Kd-tree consisting of N nodes is $O(\log_2 N)$.

Further, in order to perform a range search [5] for a given query iris Q with a distance r, it determines all iris images having euclidean distance from Q less than or equal to r. The average cost to perform a range search in Kd-tree consisting of N nodes is $O(k.N^{1-1/k})$ [3].

2.2 Speeded Up Robust Feature Transform

Speeded-Up Robust Features (SURF)[1, 2] is feature extraction algorithm which is a rotation-invariant interest point detector and descriptor. The feature points which are extracted from the images are highly distinctive and invariant. It is found to be more robust to the images having change in view, illumination, scale and occlusion. It has been used as feature representation in many applications such as object recognition [6], robot navigation [12] etc. Following are the two major steps which are followed to compute SURF features of an image.

2.2.1 Key-Point Detector

The key-points which are salient feature points in the image are identified by SURF. For this it makes use of hessian matrix. The hessian matrix $H(P, \sigma)$, at scale σ, for a given point $P(x, y)$ in an image I' is defined as follows:

$$H(P,\sigma) = \begin{bmatrix} L_{xx}(P,\sigma) & L_{xy}(P,\sigma) \\ L_{yx}(P,\sigma) & L_{yy}(P,\sigma) \\ \end{bmatrix}$$

where $L_{xx}(P,\sigma)$, $L_{xy}(P,\sigma)$, $L_{yx}(P,\sigma)$ and $L_{yy}(P,\sigma)$ are the convolution of the Gaussian second order derivatives $\frac{\partial^2}{\partial x^2}g(\sigma)$, $\frac{\partial^2}{\partial x \partial y}g(\sigma)$, $\frac{\partial^2}{\partial y \partial x}g(\sigma)$ and $\frac{\partial^2}{\partial y^2}g(\sigma)$ with the image I' at point P respectively. Second order Gaussian derivatives in Hessian matrix are approximated to speedup the computation using box filters. Convolution of images with the box filter is made fast by making the use of integral images. In order to detect the key-points at different scales, scale space representation of the image is obtained by convolving it with the box filters. In this, rather than iteratively reducing the image size, the scale space is analysed by up-scaling the filter size. In order to localize interest points in the image and over scales, non-maximum suppression in a $3 \times 3 \times 3$ neighborhood is implemented.

2.2.2 Key-Point Descriptor

Key-point descriptor is of two step process. In the first step, for each detected key-points a circular region is considered and Haar wavelet responses dx and dy in horizontal and vertical directions are computed. These responses are used to obtain the dominant orientation in the circular region which is used to generate the key-point descriptor. Feature vectors are measured relative to the dominant orientation resulting the generated vectors invariant to image rotation. In the second step, it considers a square region around each key-point and aligns it along the dominant orientation. The square region is divided into sixteen sub-regions in 4×4 format and Haar wavelet responses are computed for each sub-region. The sum of the wavelet responses in horizontal and vertical directions for each sub-region are used as key-descriptor. In order to obtain the information about the polarity of the image intensity changes the absolute values of responses are also summed. Thus, the key-descriptor V_i for i^{th} sub-region is given by

$$V_i = \{\Sigma dx, \Sigma dy, \Sigma|dx|, \Sigma|dy|\}$$

The SURF feature vector of a key-point is obtained by concatenating key-descriptor (V_i) from all sixteen sub-regions around the key-point which results a SURF feature vector of length 64. Extended version of SURF (known as SURF-128), which is more distinctive, adds couple of similar features. It uses the sums same as described above, but splits these values up further. The sums of d_x and $|d_x|$ are computed separately for $d_y < 0$ and $d_y \geq 0$. Similarly, the sums of d_y and $|d_y|$ are split up according to the sign of d_x, and as a result it doubles the number of features. Iris retrieval technique proposed in this paper uses SURF-128 (hence forth referred as SURF) for texture feature representation.

2.2.3 Matching

SURF matching is based on the nearest neighbor ratio method. For a given test image having key-points, the best candidate match image is found by identifying its nearest neighbor in the key-points. The nearest neighbors between the key-points are defined with minimum Euclidean distance from the given key-descriptor around the key-points. The probability that a match is correct is determined by computing the ratio of distance from the closest neighbor to the distance of the second closest. The distance ratio which is less than a threshold (τ) are considered.

3. PROPOSED INDEXING SCHEME

In iris based biometric identification, features which are extracted from each iris image, are mapped in to a feature space and is defined by a point $(f_1, ..., f_n)$ in the space. A metric can be defined on the space, and identification is done by finding the k nearest points in feature space to the given query point. All these feature points are indexed in such a way that it avoids the comparison with all feature points in the space while searching.

In order to identify a subject using iris color, one can use a color histogram based method. A histogram can be represented by a vector $H = [h_1, ..., h_n]$ in which each element h_j contains the number of pixels having the color j in the image and can be considered the probability density function (pdf) of the color values. Each of the different bins of the histogram can be considered as a different feature for recognition. For the given segmented iris region I, a histogram H is generated and is stored in the database. During identification for a query iris image, its nearest neighbors are found by comparing all the histogram stored in the database using the sum of the squared differences. However, several issues may raise with color histogram based method. One of the major issue is the high dimensionality of the color histograms. It

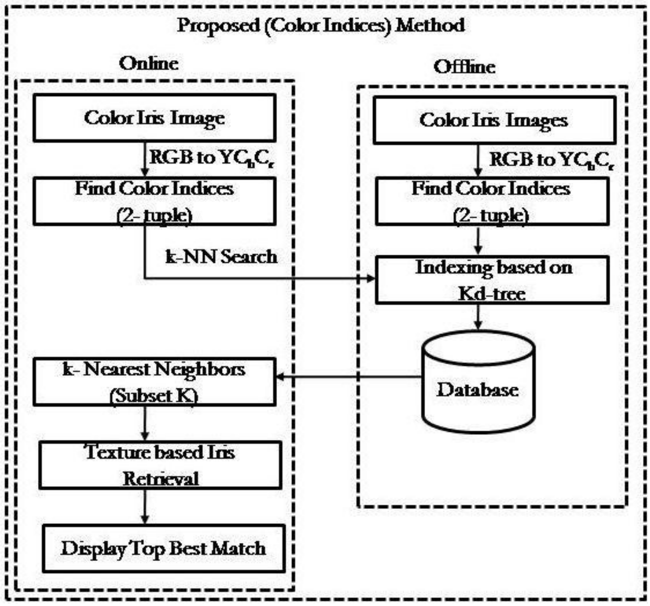


Figure 2: Overview of Proposed Technique

can be observed that even if one uses feature reduction, pre-filtering and quantization methods, the feature space still occupies more than 100 dimensions. As a result, it is difficult to index such a high dimensional features. This larger dimension also increases the computational complexity.

This section proposes Kd-tree based indexing scheme for iris database. Fig. 2 shows the overview of the proposed technique. In the proposed indexing scheme, two dimensional data are blue and red indices of iris images computed under YC_bC_r color space. Let I be the segmented iris region for any given iris image. The iris region I which is originally in RGB color space can be converted to YC_bC_r color space using the standard procedure as

$$\begin{bmatrix} Y(x,y) \\ C_b(x,y) \\ C_r(x,y) \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \frac{1}{256} \begin{bmatrix} M \end{bmatrix} \cdot \begin{bmatrix} R(x,y) \\ G(x,y) \\ B(x,y) \end{bmatrix}$$

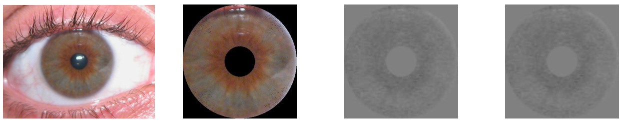
where

$$M = \begin{bmatrix} 65.73 & 129.05 & 25.06 \\ -37.94 & -74.49 & 112.43 \\ 112.43 & -94.15 & -18.28 \end{bmatrix}$$

and $R(x,y), G(x,y)$ and $B(x,y)$ are the red, green and blue pixel values of (x,y) over RGB color space respectively. Then blue and red indices of I under YC_bC_r color space can be obtained by

$$b_I = \frac{1}{|I|} \sum_{\forall(x,y)\in I} C_b(x,y), \;\; r_I = \frac{1}{|I|} \sum_{\forall(x,y)\in I} C_r(x,y) \qquad (2)$$

where $C_b(x,y)$ and $C_r(x,y)$ are the chrominance of blue and red color respectively for pixel (x,y) of I. Importantly, since YC_bC_r color space separates illumination from color, conversion from RGB to YC_bC_r color space allows the color indices to operate well in varying light levels. As a result, high indexing performance may be achieved. It can be noted that computing color indices by converting RGB color space into



(a) (b) (c) (d)

Figure 3: Computing the Color Indices (a) Input Iris Image, (b) Segmented Iris, (c) Blue Index, (d) Red Index.

rg-chromaticity color space or in the RGB color space itself may not achieve high indexing performance due to variation in illumination in the images. An example of color indices computed on segmented iris region I is shown in Fig. 3.

Let $DB = \{D_1, D_2, ..., D_N\}$ be the database of N iris images where $D_L, L = 1, 2..., N$, is a 2-tuple containing blue and red indices of the iris region I over YC_bC_r color space, *i.e*, $D_L = (b_L, r_L)$. Given the database DB, one can create a Kd-tree by inserting blue and red indices (b_L, r_L). For a given query iris image q, blue and red indices are computed over YC_bC_r color space and a subset K of k iris images which are nearest to the query image q has been found. The subset K contains all iris images satisfying $(\forall i \in K)$,

$$\|q - i\| \leq \|q - n\|, \forall n \in (DB - K) \quad (3)$$

where $\|.\|$ is a distance measure.

4. PROPOSED RETRIEVAL TECHNIQUE

The iris images obtained in the subset K can be arranged based on the Euclidean distance of color indices of these images with the query image. However, the color indices which are the average of red and blue color values of an iris image may not get the query's corresponding identity in the top best match. So in order to get the query's corresponding identity from the subset K in the top best match, iris texture is used.

This section discusses the technique to retrieve the top best match from the subset K based on the iris texture patterns. The texture patterns of iris region I which have richer information can help to improve the matching performance. Such an improvement of iris recognition system has happened due to integration of a local feature descriptor named SURF which has explained in Section 2. SURF has more discriminative power than any other local feature descriptor such as SIFT [9]. Local features are extracted by finding the key points in an image and forming key-descriptor around each detected key-point. The key-descriptor of all key-points forms the feature vector F_J, of J^{th} iris image in the subset K where $J = 1, 2, ..., k$. Finally, query image q is matched with all images in the subset K using these key-descriptor. The image which has the maximum matching points is displayed as the top best match for the given query image. Fig. 4 shows detected SURF key points of an iris region I of UPOL database.

5. PERFORMANCE EVALUATION

To determine the performance of the proposed indexing scheme, four measures, namely, *hitrate*, *bin-miss rate*, *penetration rate*, and *Cumulative Match Characteristic curve* are used.

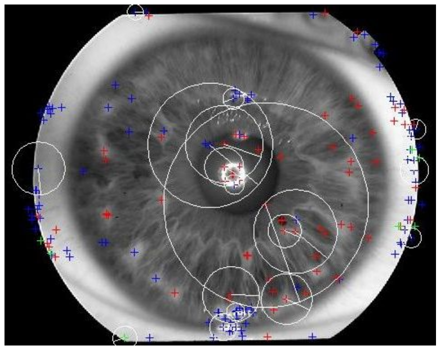


Figure 4: Detected Keypoints of Iris Image

- The *hitrate* (H_r) is the ratio of the number of times (X) that the corresponding identity has been found in the top best match to the total number of attempts made (L), *i.e.*,

$$H_r = \left(\frac{X}{L}\right) \times 100\% \quad (4)$$

- The *bin-miss rate* (B_r) is defined as $B_r = 100 - H_r$
- The *penetration rate* (P_r) is defined as the ratio of average number of images retrieved $(|K| = k)$ from the database against the query image to the total database size (N).

$$P_r = \left(\frac{1}{X}\sum_{i=1}^{X}\frac{k}{N}\right) \times 100\% \quad (5)$$

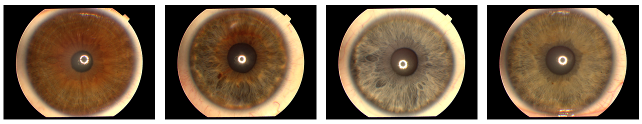
In our experiment, the number of iris images retrieved is fixed for all query images and hence it can be defined as the ratio of the subset size $(|K|)$ to the database size (N), and is given by

$$P_r = \left(\frac{|K|}{N}\right) \times 100\% \quad (6)$$

- *Cumulative Match Characteristic (CMC) curve* represents the relationship between number of correctly matched images (X) against top positions.

5.1 Experiment 1: UPOL Database

The UPOL database [10] contains 384 iris images collected from 64 subjects. It contains 3 left and 3 right iris images of each subject. Four samples of iris images from this database are shown in Fig. 5. For each subject among 3 left/right iris images, 2 images are considered for training and the remaining one image is used for testing.



(a) (b) (c) (d)

Figure 5: Iris Image Samples from UPOL Database

The blue and the red indices over YC_bC_r color space are computed for all images. It has been observed that variation

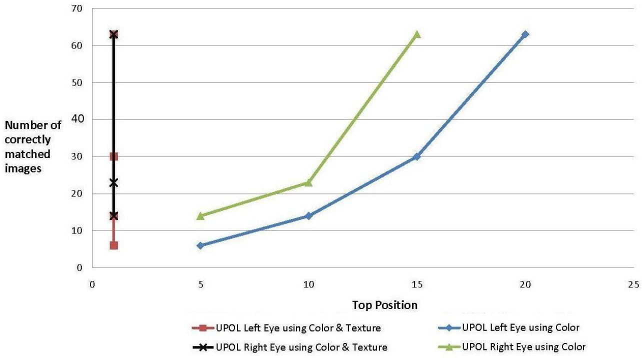


Figure 6: ***Cumulative Match Characteristic*** **Curve for UPOL Database**

of each such color index for different images of the same subject is very small while that between two subjects is large for blue and red indices. This shows that 2-tuple consisting of blue and red indices over the YC_bC_r color space can play an important role to determine a good index for each iris image. Once indexing has been done based on iris color indices (blue and red), a small subset K from the database is obtained for a query q. Testing has been carried out for various size of subset K. Table 1 shows *hitrate*, *bin-miss rate* and *penetration rate* for the proposed indexing scheme for various subset sizes. It has been found that the subsets of sizes 15 and 20 are sufficient to achieve 100% *hitrate* with 11.71% and 15.62% *penetration rate* for both right and left eye databases respectively. In the next stage, texture based iris retrieval technique is used to get the correct identity in the top best match.

UPOL Database						
Subset Size k	Left(Q=63, N=128)			Right(Q=63, N=128)		
	H_r %	B_r %	P_r %	H_r %	B_r %	P_r %
5	82.53	17.47	3.90	85.71	14.29	3.90
10	93.65	6.35	7.81	95.23	4.77	7.81
15	98.41	1.59	11.71	100	0	11.71
20	100	0	15.62	100	0	15.62

Table 1: Hitrate, Bin-miss Rate and Penetration Rate against Different Subset Size for Top Best Match

Fig. 6 shows *CMC curve* for UPOL left and right eye databases using only color as well as using color and texture. It can be observed that the number of correctly matched images lie in between top 5 to 20 positions when color is used for left eye. Similarly it lie in between top 5 to 15 positions for right eye (refer Table 1). But it lie always in first position for both database when color and texture are used. Hence, the combination of iris color and texture is the best choice to achieve higher identification accuracy.

5.2 Experiment 2: Noisy UBIRIS Database

The performance of the proposed indexing scheme is also evaluated on noisy UBIRIS [13] database and analysed against the group based color indexing scheme proposed in [14]. UBIRIS database consists of 1860 iris images collected from 372 subjects (5 samples each) in two sessions. For Session 1, images have been captured from 241 persons in controlled environment in order to mimize noise factors, specially those relative to reflections, luminosity and contrast. For Session 2, images are captured from 131 persons by changing the capture location to introduce natural luminosity factor. This has enabled the appearance of heterogeneous images with respect to reflections, contrast, luminosity and focus problems. All the images from both sessions have been classified with respect to three parameters. The classification statistics are given in Table 2.

Parameter	Good	Average	Bad
Focus	73.83 %	17.53 %	8.63 %
Reflections	58.87%	36.78%	4.34%
Visible Iris	36.73%	47.83%	15.44 %

Table 2: Classification of Images with respect to Three Parameters

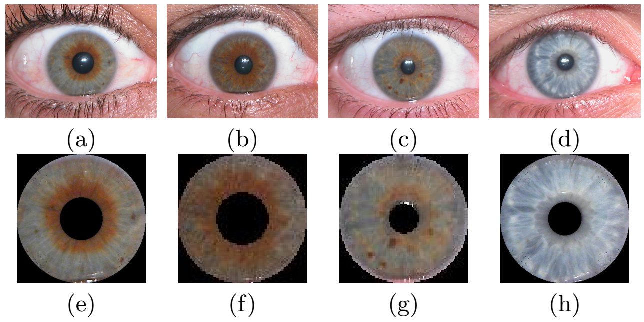


Figure 7: Good Visible Iris Samples from UBIRIS Database (Segmentation Success Cases)

Some images for both good visible iris and bad visible iris along with the their segmented iris are shown in Fig.7 and Fig.8 respectively. For noisy UBIRIS database, iris region (I) from the eyelid has been segmented separately using the method discussed in [8]. The experiment has been conducted for the UBIRIS database after removing the noisy images as shown in Fig. 8 where the proper iris segmentation does not happen because of much occlusion present in the eye. If one considers only those subjects having at least two good segmented iris regions (I), one can get a total of 237 out of 241 different subjects from Session 1. Similarly from Session 2, a total of 130 out of 131 different subjects are considered. For each subject, one image is taken for training and remaining images are considered for testing. Then color indices are computed for each segmented iris region (I). In this case, the variation in color indices between two images of the same subject is found to be significant.

In group based color indexing scheme, a subject is considered for indexing if it contains at least two well segmented iris images. A total of 238 subjects are considered for indexing. In each subject, one image is taken for training and rest of the images are considered for testing. Table 3 shows *hitrate*, *bin-miss rate* and *penetration rate* of the proposed indexing scheme against the group based color indexing scheme for various subset k and search range n_{range} respectively. It can be observed that in the proposed indexing scheme almost the same *hitrate* is achieved with less

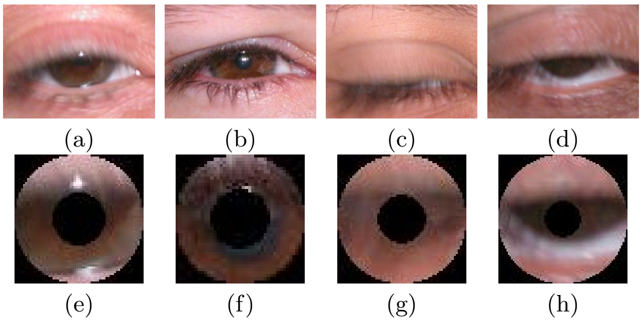


Figure 8: Bad Visible Iris Samples from UBIRIS Database (Segmentation Failed Cases)

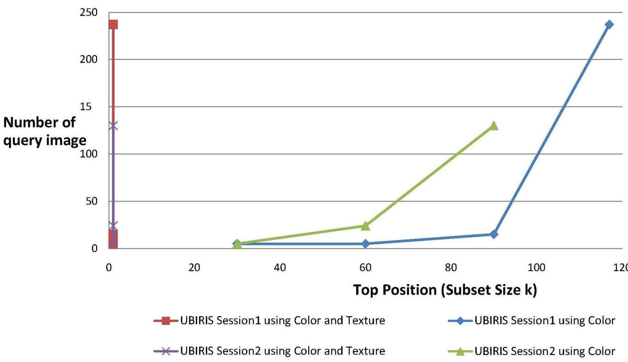


Figure 9: ***Cumulative Match Characteristic*** **Curve for UBIRIS Database**

penetration rate as compared to the group based color indexing scheme. Also in group based color indexing scheme the *penetration rate* increases drastically as n_{range} increases. Further, 100% *hitrate* is achieved with only 49.36% *penetration rate* by the proposed indexing scheme for Session 1 while 99.78% of *hitrate* is achieved with 62.22% of *penetration rate* in the group based color indexing scheme proposed in [14]. Similarly, for Session 2, 100% *hitrate* is achieved with 37.97% of *penetration rate* which is very less as compared to 99.79% *hitrate* with 51.58% *penetration rate* obtained in [14].

Fig. 9 shows *CMC curve* for UBIRIS Session 1 and Session 2 databases using only color as well as using color and texture. It can be observed that the number of correctly matched images lie in between top 30 to 120 positions when color is used for Session 1. Similarly, it lie in between top 30 to 90 position for Session 2 (refer Table 3). As like in UPOL database, it lie always in first position for both database when color and texture are used.

5.2.1 Robust to Change in Gaze, Illumination, Partial Occlusion & Scale

Experiments have been performed to test the sensitivity of the proposed indexing scheme due to change in gaze, illumination, partial occlusion and scale. Fig. 10 shows sample images from UBIRIS database with change in gaze, illumination, partial occlusion and scale along with its corresponding segmented iris region considered for the experiment. The experiment shows that the proposed indexing scheme is capable of indexing a large iris database that is eliminating most

I. Proposed Indexing Scheme, UBIRIS Database						
Subset Size k	S-1 (Q=912, N=237)			S-2 (Q=491, N=130)		
	H_r %	B_r %	P_r %	H_r %	B_r %	P_r %
30	94.93	5.07	12.65	92.95	7.05	12.65
40	95.78	4.22	16.87	96.78	3.22	16.87
50	97.04	2.96	21.09	98.19	1.81	21.09
55	98.31	1.69	23.20	98.27	1.73	23.20
60	99.73	0.27	25.31	99.89	0.11	25.31
65	99.73	0.27	27.42	99.89	0.11	27.42
90	99.73	0.27	37.97	100	0	37.97
117	100	0	49.36	100	0	49.36
II. GBCIS Reproduced From [14], UBIRIS Database						
n_{range}	S-1 (Q=893, N=238)			S-2 (Q=498, N=130)		
	H_r %	B_r %	P_r %	H_r %	B_r %	P_r %
0	44.90	55.10	1.70	44.37	55.63	1.65
1	95.18	4.82	10.43	92.77	7.23	6.94
2	98.54	1.46	24.44	98.79	1.21	15.93
3	99.66	0.34	38.92	99.19	0.81	27.75
4	99.78	0.22	51.78	99.39	0.61	39.99
5	99.78	0.22	62.22	99.79	0.21	51.58

Table 3: Comparison Showing Hitrate, Bin-miss Rate and Penetration Rate Against Different Subset Size for Top Best Match

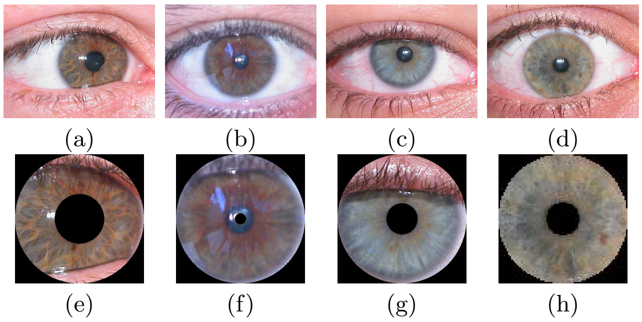


Figure 10: UBIRIS Iris Images (a) Change in Gaze, (b) Illumination, (c) Partial Occlusion, (d) Different Scale (200×150), (a)-(c) 800×600, (e)-(h) Its Corresponding Segmented Region.

of the possible matches only with 2-tuple color indices. It has been observed that recognition accuracy is fairly insensitive to change in gaze, partial occlusion and illumination. Hence the proposed indexing scheme increases the robustness of iris recognition system.

Further, a test has been conducted to show that the proposed indexing scheme is scale invariant. For that, images of various scales are considered for testing and the results are tabulated in Table 4 for the subset of sizes 20 and 50 for UPOL and UBIRIS respectively. It could be observed that *hitrates* of the proposed indexing scheme remains almost same even though there is a change in the scale and this is true for other subset size k as well. In general, color indices are the best choice for indexing as they are invariant to translation and rotation about the viewing axis and change slowly under change of angle of view.

5.3 Selection of Subset

Determining the size of subset $|K|$ in an important factor towards achieving the desired performance and efficiency.

Scales	UPOL (k=20)		UBIRIS (k=50)	
	L (H_r)	R (H_r)	S-1 (H_r)	S-2 (H_r)
800 × 600	100	100	97.04	98.19
650 × 450	100	100	97.04	98.19
400 × 300	100	100	96.12	97.01
300 × 200	100	100	96.12	97.01
200 × 150	100	100	96.12	97.01

Table 4: Hitrate in % Against Various Scaled Images for Subset Size of 20 and 50 for Top Best Match.

Ideally, one needs a subset of size such that 100% *hitrate* is achieved with low *penetration rate*. However in practice, increase in the subset size to achieve 100% *hitrate* also increases the *penetration rate*. So in our experiments, an optimum size of the subset to achieve near 100% *hitrate* is with minimum *penetration rate*. The selection of such an optimum value is dependent on the database size, image quality etc and may differ for different databases. It can be noted from Table 1 and Table 3 for UPOL database, the subset size varies between 5 and 20 where as for UBIRIS database between 30 and 117 to get the desired results.

5.4 Time Complexity

In the proposed indexing scheme Kd-tree is constructed for N iris images, each having 2-tuple color indices. It uses linear median-finding algorithm to compute the median at each level to partition the space and it takes $O(NlogN)$ time complexity to construct the tree. And for a given iris query image, Kd-tree uses k-NN search in order to produce subset size of k. It is clear that on an average $O(logN)$ inspections are necessary because any nearest neighbor search requires to traverse at least one leaf of the tree. During iris retrieval, one to one match between query and all images in subset of size k is performed. Hence the total time taken to display the top best match is $O(logN + k)$ where k is the subset size (i.e the number of images compared during iris retrieval which is constant).

6. CONCLUSIONS

This paper proposes an efficient retrieval technique which uses a new indexing scheme for iris biometric database and has shown its effectiveness to reduce the search space. It has been shown that during iris indexing, the value of color indices does not change even with the change in gaze, illumination, partial occlusion and scale which makes the proposed indexing scheme superior to other indexing schemes. Also SURF local descriptor which is used for iris retrieval to get the correct identity in the top best match further enhances the performance of the iris recognition system. The proposed indexing scheme is compared with group based color indexing scheme proposed in [14] and it is found better performance with respect to *hitrate* and *penetration rate*. A 100% *hitrate* is achieved with only 49.36% *penetration rate* by the proposed indexing scheme for Session 1 while 99.78% of *hitrate* is achieved with 62.22% of *penetration rate* in the group based color indexing scheme proposed in [14]. Similarly for Session 2, 100% *hitrate* is achieved with 37.97% of *penetration rate* which is very less as compared to 99.79% *hitrate* with 51.58% *penetration rate* obtained in [14]. The proposed indexing scheme is found to be robust as it is invariant to change in gaze, illumination, partial occlusion and scale.

7. ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their constructive feedback.

8. REFERENCES

[1] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[2] H. Bay, T. Tuytelaars, and L. V. Gool. SURF: Speeded Up Robust Features. In *ECCV '06: Proceeding of 9th European Conference on Computer Vision*, pages 404–417, 2006.

[3] J. L. Bentley. Multidimensional Binary Search Trees Used for Associative Searching. *Communications of the ACM*, 18(9):509–517, 1975.

[4] J. L. Bentley. K-d Trees for Semidynamic Point Sets. In *SCG '90: Proceedings of the sixth annual symposium on Computational geometry*, pages 187–197, 1990.

[5] J. L. Bentley and J. H. Friedman. Data Structure for Range Searching. *ACM Computing Surveys*, 11:397–409., 1979.

[6] B. Fasel and L. Van Gool. Interactive museum guide: accurate retrieval of object descriptions. In *ICAMR '07: Proceedings of the 4th international conference on Adaptive multimedia retrieval*, pages 179–191, 2007.

[7] J. Fu, H. J. Caulfield, S.-M. Yoo, and V. Atluri. Use of Artificial Color Filtering to Improve Iris Recognition and Searching. *Pattern Recognition Letters*, 26(14):2244–2251, 2005.

[8] C. Kimme, D. Ballard, and J. Sklansky. Finding Circles by an Array of Accumulators. *Communication of the ACM*, 18(2):120–122, 1975.

[9] D. Lowe. Object Recognition from Local Scale-Invariant Features. In *ICCV '99: Proceeding of IEEE 7th International Conference on Computer Vision*, pages 1150–1157, 1999.

[10] Michal DobeŽ and Libor Machala. Iris Database. *http://www.inf.upol.cz/iris/*.

[11] R. Mukherjee and A. Ross. Indexing Iris Images. In *ICPR '08: Proceeding of International Conference on Pattern Recognition*, 2008.

[12] A. C. Murillo, J. J. Guerrero, and C. Sagüés. SURF Features for Efficient Robot Localization with Omnidirectional Images. In *ICRA '07: Proceeding of International Conference on Robotics and Automation*, pages 3901–3907, 2007.

[13] Proenca.H and Alexandre. UBIRIS: A Noisy Iris Image Database. *Lecture Notes in Computer Science*, 3617:970–977, 2005.

[14] N. B. Puhan and N. Sudha. A Novel Iris Database Indexing Method using the Iris Color. In *ICIEA '08: Proceeding of IEEE Conference on Industrial Electronics and Applications*, pages 1886–1891, 2008.

[15] H. Samet. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, 1990.

[16] 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*, 38(4):1021–35, 2008.

[17] L. Yu, D. Zhang, K. Wang, and W. Yang. Coarse Iris Classification Using Box-Counting to Estimate Fractal Dimensions. *Pattern Recognition*, 38(11):1791–1798, 2005.