Indexing Multimodal Biometric Databases Using Kd-Tree with Feature Level Fusion

Umarani Jayaraman, Surya Prakash, and Phalguni Gupta

Department of Computer Science and Engineering,
Indian Institute of Technology Kanpur, Kanpur-208 016, India
{umarani,psurya,pg}@cse.iitk.ac.in

Abstract. This paper proposes an efficient indexing technique that can be used in an identification system with large multimodal biometric databases. The proposed technique is based on Kd-tree with feature level fusion which uses the multi- dimensional feature vector. A multi dimensional feature vector of each trait is first normalized and then, it is projected to a lower dimensional feature space. The reduced dimensional feature vectors are fused at feature level and the fused feature vectors are used to index the database by forming Kd-tree. The proposed method reduces the data retrieval time along with possible error rates. The system is tested on multimodal databases (feature level fusion of ear, face, iris and signature) consists of 5400 images of 150 subjects (*i.e.* 9 images per subject per trait). Out of the 9, 8 images are used for training and 1 is used for testing. The performance of the proposed indexing technique has been compared with indexing based on score level fusion. It is found that proposed technique based on feature level fusion performs better than score level fusion.

Keywords: indexing, feature level fusion, Kd-tree, multi-dimensional data structure.

1 Introduction

Biometric system provides an automated method to verify or to identify an individual based on unique behavioral or physiological characteristics. Such a system consists of biometric readers, or sensors; feature extractor to compute salient attributes from the input template; and feature matcher for comparing two sets of biometric features. An authentication system consists of two subsystems; one for enrollment while other for authentication. During enrollment, biometric measurements are captured from a subject and relevant information is stored in the database. The task of the authentication module is to recognize a subject as a later stage, and is either identification of one person among many, or verification that a person's biometric matches a claimed identity [1]. Let T_i be the feature vector of i^{th} template in the database of d dimension and is defined as follows:

$$T_i = [f_1, ..., f_d] \quad (1)$$

R. Sekar and A.K. Pujari (Eds.): ICISS 2008, LNCS 5352, pp. 221–234, 2008.

If the query image Q with feature vector of d dimension is defined as $Q = [q_1, ...q_d]$, then $\forall j$, q_j may not be same as f_j, where f_j and q_j are the j^{th} feature values of T_i and Q respectively. For a given query template Q, the problem of identification system is to find the n nearest neighbors in the database consisting of N templates. To make the system more powerful and fast, a feature matcher should search for the templates with some pruning technique. Traditional databases index the records in an alphabetical or numeric order for efficient retrieval. In biometric database, there is no natural order by which one can keep the biometric templates. Hence there is a need for good indexing technique to make the search process more efficient. Since the feature vector generated for every templates are known a priori, these vectors can be arranged in such a way that an efficient searching algorithm can be used.

In the literature, there are few techniques available to index the biometric templates. These techniques either cluster the multi-dimensional templates and uses cluster information for searching or map the multi-dimensional feature vector to a scaler value [2] and use an existing data structure to index it. The study of reducing the search space of biometric databases is already made by Center for Unified Biometrics and Sensors (CUBS) group. Their effort is based on the binning/clustering technique. The effort of binning was performed in [3] and it was demonstrated that the search space could be reduced to approximately 5% of the original database. The data was clustered using K-means clustering algorithm. The test template is then associated with the clusters it is closest to, with the new search space being the templates within these closest clusters. However, the binning approach needs re-partition the entire database on addition of new samples to it. Thus binning/clustering systems are useful only in cases of static databases. In [2], CUBS group made another effort to index biometrics database using pyramid technique. Pyramid technique partitions a d-dimensional universe into $2d$ pyramids that meet at the center of the universe. Then, every pyramid is divided into several slices parallel to the basis of the pyramid. The d-dimensional vectors representing points in space are approximated by one-dimensional quantities, called pyramid values, which are indexed by a B+ tree. However, the pyramid technique is ineffective in situations when the data points fall on or near the boundaries of the original space. Another problem in this technique is the increased storage overhead due to the need to keep the index of both the pyramid values and the original points. Kd-tree stores actual data (original points) based on the i^th discriminator as a key value in a node, and hence there is no storage overhead as like pyramid technique. In addition since Kd-tree is based on space partitioning indexing technique, there is no overlapping between nodes as like bounding region based indexing techniques such as R-tree, R*-tree, M-tree and X-tree [4,5]. This paper proposes an efficient indexing technique which reduces the search space of multimodal biometric databases for any query template using Kd-tree with feature level fusion.

Section 2 presents dimension reduction technique and Kd-tree which serve as the foundation for further discussion. Section 3 describes the feature extraction algorithms for Iris, Signature, Ear and Face biometric traits. In Section 4 the indexing technique has been proposed using Kd-tree with feature level fusion for the multimodal identification system. Results are analysed in the section 5. Conclusion is given in Section 6.

2 Preliminaries

This section discusses some of the basic techniques which are required in developing the proposed indexing technique. Section 2.1 describes dimension reduction technique that has been used to reduce the dimension of the feature vectors. Section 2.2 describes the Kd-tree data structure which is used to index the databases by forming Kd-tree.

2.1 Dimension Reduction Technique

Suppose, we have a dataset T consisting of $\{t_i\}_{i=1}^{N}$ training samples. Each sample is described by a set of features $F(d = |F|)$, so there are N samples described by d dimensional feature vector each. This can be represented by feature object matrix $T_{d\times N}$ where each column represents a sample. The objective of dimension reduction is to reduce the data into another set of features F' where $k = |F'|$ and $k < d$; $T_{d\times N}$ is reduced to $T'_{k\times N}$. Typically, this is a linear transformation $T' = WT$ that reduces dataset T to T' in k dimension, where $W = W_{k\times d}$ is the transformation technique. Principal Component Analysis (PCA) which is the dominant dimension reduction technique, transforms the data into a reduced space that captures most of the variance in the data. PCA reduces dimension of a dataset by retaining those characteristics of the dataset that contribute most to its variance. It can be done by keeping lower-order principal components and ignoring higher-order ones. Such low-order components often contain the most important aspects of the data. PCA can be mathematically defined [6] as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie

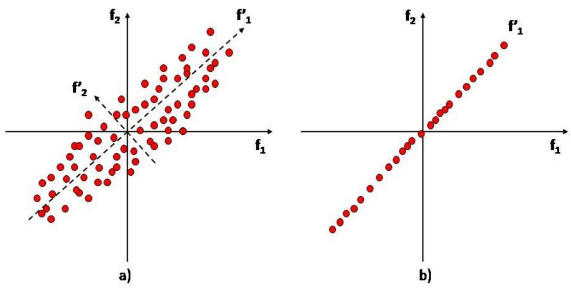


Fig. 1. PCA in $2-D$, the variance in the f'_1 direction is maximum. a) Solid lines: The original basis; Dashed lines: The PCA basis; b) The projection (1D reconstruction) of the data using the first principal component.

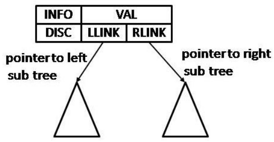


Fig. 2. Structure of Kd-tree node

on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on. PCA is theoretically an optimum transformation for a given data in least square terms. Dimensionality reduction in PCA is accomplished by computing a set of eigenvalues of total scatter matrix S_t of the data samples defined as:

$$S_t = \sum_{i=1}^{N} (t_i - m)(t_i - m)^T \tag{2}$$

where m is the mean value of the sample set T. For dimensionality reduction, k (where $k < d$) eigenvectors $U = [u_1, ..., u_k]$ corresponding to first k largest eigenvalues of S_t are selected. Reduced dimension training samples $T' = [t'_1, ..., t'_n]$ can be obtained by the transformation $T' = U^T T$. Now, when a probe template t_p is presented for identification/verification, it is projected on U to obtain a reduced dimension vector $t'_p = U^T t_p$. Geometric interpretation of PCA is shown in Fig. 1. To see how the data is spread, we encapsulate the dataset inside an ellipse and take a look at the major and minor axes that form the vectors f'_1 and f'_2. These are the principal component axes *i.e.* the base vectors that are ordered by the variance of the data. PCA finds these vectors and gives a $[f'_1, f'_2] \rightarrow [f'_1]$ transformation. While this example is for $2 - D$, PCA works for multi-dimensional data, and it is generally used with high dimensionality problems.

2.2 Kd-Tree Data Structure

The proposed indexing technique is based on the Kd-tree data structure [7,4,8]. 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. Structure of a node in Kd-tree is given in Fig. 2. 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 $level(P) \bmod k$. 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

$$NEXTDISC(i) = (i+1) \bmod k \tag{3}$$

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 [7] 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 [9] for a given query template Q with a distance r, it determines all templates T 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 $O(k.N^{1-1/k})$ [7].

3 Feature Extraction

A template in the proposed system is represented by four real feature vectors: iris feature vector T_I, signature feature vector T_S, ear feature vector T_E and face feature vector T_F. The signature feature vector is obtained using parameter extraction algorithm [10,11], while other feature vectors are extracted using discrete haar wavelet transform. In this section, the process of feature extraction for each trait has been discussed.

3.1 Iris Feature Extraction

Features are extracted from the iris image using localization of inner and outer iris boundaries [12,13]. In the proposed strategy, the inner iris boundary is localized on the iris image using circular hough transformation [14,15]. Once the inner iris boundary (which is also the boundary of the pupil) is obtained, outer iris is determined using intensity variation approach [16]. The annular portion of iris after localization is transformed into rectangular block to take into consideration the possibility of pupil dilation. This transformed block is used for feature extraction using Discrete Haar Wavelet Transform (DHWT). Haar wavelet operates on data by calculating the sums and differences of adjacent values. It operates first on adjacent horizontal values and then on adjacent vertical values. The decomposition is applied up to four levels on transformed rectangular iris block as shown in Fig. 3. A d dimensional real feature vector T_I is obtained from the fourth level decomposition and is given by

$$T_I = [i_1, ..., \ i_{d_I}] \tag{4}$$

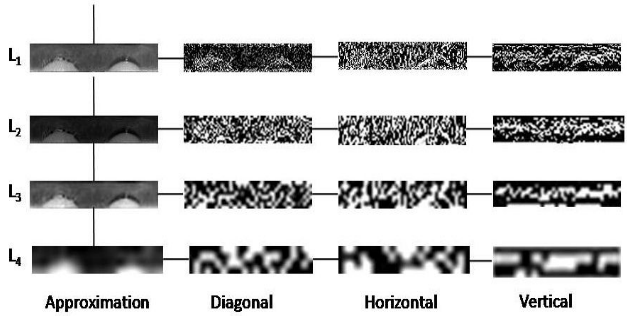


Fig. 3. Four levels discrete haar wavelet transform on iris strip

3.2 Signature Feature Extraction

The signature image is scanned and cropped [approximately 300 × 150 pixels] in the required format. The image is first passed to the noise removal module which returns a noise free image in the form of a $2-D$ integer array of same size as the input image. The image extraction module is called to extract the binary, high pressure region (HPR), and thinned images from the array.

Features can be classified into two types: global and local features, where global features are characteristics that identify or describe the signature as a whole and local features are confined to a limited portion (e.g. a grid) of the signature [10,11]. Examples of global features include width and height of individual signature components (as shown in Fig. 4.), width to height ratio, total area of black pixels in the binary and high pressure region (HPR) images, horizontal and vertical projections of signature images, baseline, baseline shift, relative position of global baseline and center of gravity with respect to width of the signature, number of cross and edge points, slant, run lengths etc. These parameters are extracted from a signature template and are stored as feature vector of d dimension as follows:

$$T_S = [s_1, ..., \ s_{d_S}] \tag{5}$$

3.3 Ear Feature Extraction

The given input side face image is first resized to 240 × 340 pixels, and the ear part is cropped from the side face image. Discrete haar wavelet decomposition

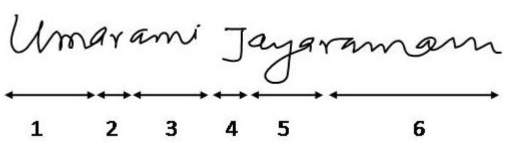
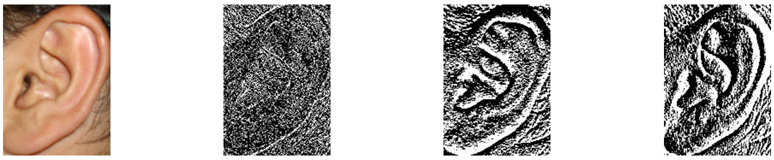


Fig. 4. Different components of the signature image



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

Fig. 5. Two levels of haar wavelet transform on ear image (a) Input ear image, (b) Diagonal coefficient, (c) Horizontal and (d) Vertical

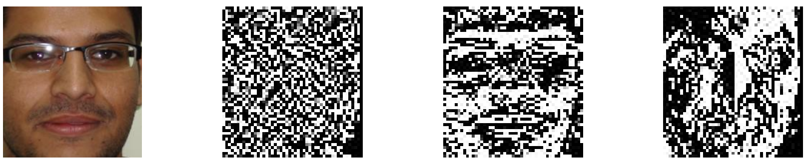
is applied to the cropped ear image to extract the wavelet Approximation (CA), Diagonal (CD), Horizontal (CH) and Vertical (CV) coefficients. Fig. 5. shows an example for the two levels of the haar wavelet decomposition applied on the cropped input ear image. The decomposition is applied up to the fifth level on cropped ear image and a d dimensional real feature vector T_E is obtained from the fifth level decomposition is stored as

$$T_E = [e_1, ..., \ e_{d_E}] \tag{6}$$

3.4 Face Feature Extraction

The given input face image is resized to 640 × 480 pixels and face part of the image is cropped leaving the background (detected face image of size 120 × 250 pixels). Haar wavelet is used for extracting the features from a detected face image. The detected input image is decomposed up to the required level giving an Approximation (CA), Vertical (CV), Horizontal (CH) and Diagonal (CD) coefficients using the haar wavelet transformation. Fig. 6. shows an example for the two levels of haar wavelet decomposition is applied on the detected face image and the corresponding Diagonal, Horizontal and Vertical coefficients. A d dimensional real feature vector T_F is obtained from the fifth level decomposition and is given by

$$T_F = [f_1, ..., \ f_{d_F}] \tag{7}$$



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

Fig. 6. Two levels of haar wavelet transform on face image (a) Input face image, (b) Diagonal coefficient, (c) Horizontal and (d) Vertical

4 The Proposed Indexing Technique

The proposed method is based on Kd-tree with feature level fusion. Feature level fusion [17] refers to combining different features sets extracted from multiple biometric sources. When the feature sets are homogeneous (e.g. multiple measurements of a person's signature), a single resultant feature vector can be calculated as a weighted average of the individual feature vector. When the feature sets are non-homogeneous (e.g, features of different biometric modalities like face and signature), we can concatenate them to form a single feature vector. The following subsection explains the concatenation of non-homogeneous feature vectors (ear, face, iris, signature) and Fig. 7. shows the overview of the proposed indexing technique.

4.1 Feature Normalization

The individual feature values of vectors $X = [x_1, x_2....x_m]$ and $Y = [y_1, y_2....y_n]$ may exhibit significant difference in range as well as form (i.e. distribution). So there is a need of feature level normalization to modify the location (mean) and scale (variance) of the features values via a transformation function in order to map them into a common domain. Adopting an appropriate normalization scheme also helps to address the problem of outliers in feature values. Let x and x' denote a feature vectors before and after normalization, respectively. The min-max technique of normalization can be used to compute x' as:

$$x' = \frac{x - min(F_x)}{max(F_x) - min(F_x)} \tag{8}$$

where F_x is the function which generates x, and $min(F_x)$ and $max(F_x)$ represent the minimum and maximum values of F_x for all possible x values, respectively. The

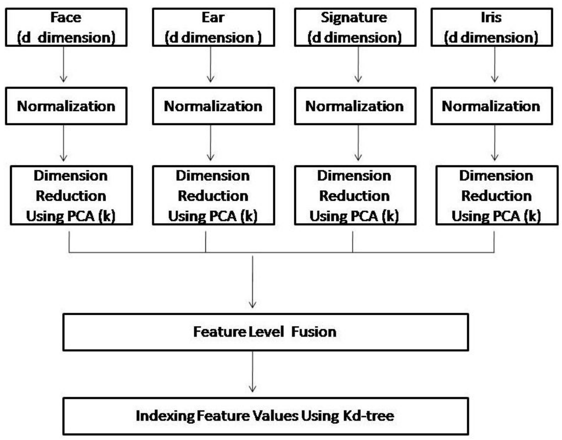


Fig. 7. Overview of proposed indexing technique

min-max technique is effective when the minimum and the maximum values of the component feature values are known beforehand. Normalizing the feature values results in modified feature vectors $X' = [x'_1, x'_2....x'_m]$ and $Y' = [y'_1, y'_2....y'_n]$. In the proposed technique, feature values for all traits are normalized between 0 and 1 using min-max technique.

4.2 Dimension Reduction Using PCA

The given real four feature vectors T_I, T_S, T_E and T_F of d dimension are reduced to k dimension using the PCA. The features extracted from four traits are of very high dimensions. Since all the features generated are not equally important, proposed technique uses PCA to reduce the dimensions which represents the compact information. In the proposed technique, the four traits features of ear, face, iris and signature are reduced from e_{d_E}, f_{d_F}, i_{d_I} and s_{d_S} to e_{k_E}, f_{k_F}, i_{k_I} and s_{k_S} dimensions respectively.

4.3 Feature Level Fusion

The reduced k dimension features are fused to yield a new feature vector T_N to represent an individual. The vector T_N of dimensionality 4k, $(4k \leq (e_{d_E} + f_{d_F} + i_{d_I} + s_{d_S}))$ can be generated by augmenting vectors T_I, T_S, T_E and T_F of iris, signature, ear and face respectively.

4.4 Indexing the Feature Vectors

The fused feature values are indexed using the Kd-tree. Kd-tree has been proposed for efficient storage and retrieval of records having multi-dimensional feature vector. Kd-tree is an appropriate data structure for biometric identification system particularly in the analysis of execution of range search algorithm. The proposed system decreases the search time as the Kd-tree is supporting the range search with good pruning.

4.5 The Technique

Indexing the database implies a logical partitioning of the data space. Based on the above steps, the algorithm for proposed indexing technique is given in Algorithm 1. In this technique, reduction makes use of Kd-tree structure for

Algorithm 1. Proposed Indexing Technique

1: Normalize the feature vectors of each trait
2: Reduce the feature vectors using PCA.
3: Fuse the feature vectors at feature level.
4: Form the Kd-tree with the fused feature vectors for indexing.
5: Invoke this indexing technique through multimodal identification system.

organizing the data in such away that each node can store one biometric template. In case of range search, only a fraction of the database lying within the range of the indices of the test template would be the search of the entire database.

5 Experimental Results

In order to demonstrate the indexing using Kd-tree for multimodal identification, data for four traits (iris, signature, ear and face) are obtained from the database available at Indian Institute of Technology, Kanpur (IITK). In the following subsections, first we briefly discuss the databases that we have used, and then present our results of indexing on multimodal identification using Kd-tree with feature level fusion and score level fusion.

Iris Database: IITK iris image database is used for the iris data. The database comprises of 1350 iris images of 150 subjects (9 images per subject) from their left eye. The images are JPEG in gray scale and are resized with 450 $\times$ 350 pixels resolution. Out of the 9, 8 images are used for training and 1 image is used for testing.

Signature Database: IITK signature database is used for signature data. The database consists of 1350 signature images of 150 subjects (9 images per subject). These images are in JPEG format. In the predefined sheet, users are asked to sign their signatures in the 9 boxes provided in the sheet. The sheet is scanned at 200 dpi as a gray scale image. Size of the scanned image is around 1700×2300 pixels and the size of one box is 300×150 pixels. Out of the 9, 8 images are used for training and 1 image is used for testing. A sample scanned signature sheet is shown in Figure 8.

Ear Database: Ear database consists of 1350 side images of human faces from IITK with the resolution of 240 $\times$ 340 pixels. The images are taken from 150 subjects (9 images per subject). These images are captured using a digital camera from a distance of 0.5 to 1 meter. Ears are cropped from these input images and used for the feature extraction.

Face Database: Face database consists of 1350 images (9 images per subject) of 150 subjects from IITK. These images are captured using a digital camera from a distance of 0.5 to 1 meter. The face is localized (120 $\times$ 250 pixels) from the given input image by clicking three points. Two points on left and right eyes

Fig. 8. Scanned signature sheet

and third point on lower lip portion. The localized face (detected face) image is used for the feature extraction.

In our experiment the dimension of feature vectors for each of the traits are found to be 64 for ear and face, 88 for the iris and 27 for the signature respectively.

5.1 Score Level Fusion Indexing Technique

In the score level fusion technique, we index the data of iris, ear, face and signature separately using four Kd-trees, one for each trait. For a given query template, identification is performed using Kd-trees of individual traits (*i.e* iris, ear, face and signature) and the results of the individual traits are fused to get the top matches. In feature level fusion there is only one Kd-tree is formed. For a given query template, identification is performed using the Kd-tree formed to get the top matches.

The experiment proceeds as follows. In our experiment, 64 dimension of face and ear as well as 88 dimension of iris, 27 dimension of signature feature values are reduced to 10 dimensions correspondingly (Top 10 eigen values are considered). This reduced feature vectors are used to index the database by forming four Kd-tree one for each trait. It takes the root of a Kd-tree and query template Q of k dimensions $[q_1, ...q_k]$ as input and retrieves all sets of templates T with k dimension $[t_1, ...t_k]$ that lies within distance r from Q. Proximity is quantified between them using Euclidean distance. The indexing technique built with Kd-tree for individual traits are invoked (ear, face, iris and signature). The distance r of each tree has to vary to get sets of matched IDs. Fix distance r as an optimum, where we get the minimum FRR in the multimodal identification system. As a result, we get four separate sets of matched IDs for ear, face, iris and signature respectively. These matched IDs are fused using weighted sum rule to declare the top matches which are the nearest matched IDs. Matching score M for an $ID\ x$ in weighted sum rule is defined as follows:

$$M = w_E \times occ_E(x) + w_F \times occ_F(x) + w_I \times occ_I(x) + w_S \times occ_S(x) \quad (9)$$

where occ_E(x), occ_F(x), occ_I(x), occ_S(x) are the occurrences of $ID\ x$ in ear, face, iris and signature result dataset respectively and w_E, w_F, w_I and w_S are respective assigned weights such that $w_E + w_F + w_I + w_S = 1$. The weights w_E, w_F, w_I and w_S are modeled using the accuracy of the verification system when only individual trait is considered and be defined as follows:

$$Acc_T = Acc_E + Acc_F + Acc_I + Acc_S \quad (10)$$

$$w_E = \frac{Acc_E}{Acc_T}, w_F = \frac{Acc_F}{Acc_T}, w_I = \frac{Acc_I}{Acc_T}, w_S = \frac{Acc_S}{Acc_T} \quad (11)$$

where $Acc_E = 96.0$, $Acc_F = 96.2$, $Acc_I = 95.5$ and $Acc_S = 86.6$ are various accuracies through our experiments we have got when they are individually used in the verification system. The matching score M is calculated for each of the matched IDs for an $ID\ x$. This matching scores are sorted to declare the top matches which are the nearest matched IDs for an $ID\ x$. In this experiment,

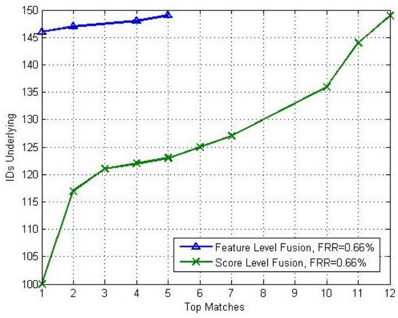


Fig. 9. Top matches vs number of IDs underlying

distance of $r_e = 17$, $r_f = 18$, $r_i = 1200$, and $r_s = 30$ the minimum FRR=0.66% is achieved, where r_e, r_f, r_i and r_s are distance of ear, face, iris, and signature respectively.

5.2 Feature Level Fusion Indexing Technique

In case of feature level fusion the fused feature vectors (ear, face, iris and signature) T_N of dimensionality $4k$ is used. In our experiment, 64 dimension of face and ear as well as 88 dimension of iris, 27 dimension of signature feature vectors are reduced to 10 dimensions correspondingly (Top 10 eigen values are considered). These reduced 10 dimensions from each of the traits are fused together, as a result we get 40 dimension fused feature vectors (T_N). This 40 dimensional feature vectors are used to index the database by forming Kd-tree. The process of identification built with Kd-tree is invoked. It takes the root of a Kd-tree and query template Q of k dimension $[q_1, ...q_k]$ as input and retrieves all sets of templates T with k dimensions $[t_1, ...t_k]$ that lies within distance r from Q. Proximity is quantified between them using Euclidean distance. As a result, we get the set of matched IDs then we sort them based on the occurrence and declare the top matched IDs. In this experiment, the distance of $r = 0.67$ the minimum $FRR = 0.66\%$ is acheived.

5.3 Comparison of Feature Level Fusion and Score Level Fusion

In score level fusion there are four separate Kd-tree is formed, one for each trait, while only one Kd-tree is formed in feature level fusion. Hence the search process is more efficient in case of feature level fusion. Fig.9 shows the top matches against IDs underlying for the 150 query templates for both score level fusion and feature level fusion. Indexing based on feature level fusion performs better than the score level fusion. Out of the 150 query template, 146 IDs fall in the first match and 147 IDs fall in the top second match and 149 IDs fall in the top 5 matches with 0.66% FRR. In score level fusion, out of the 150 IDs, 100 IDs fall in the first match, 117 IDs fall in the top second match and 149 IDs fall in the top 12 matches with 0.66% FRR.

6 Conclusion

This paper has proposed an efficient indexing technique using Kd-tree with feature level fusion to reduce the search space of large multimodal biometric databases. Kd-tree has been used for efficient storage and retrieval of records having multi-dimensional feature vector. Kd-tree is shown as appropriate data structure for biometric identification system particularly in the analysis of execution of range search algorithm. The proposed technique decreases the search time as only one Kd-tree is formed which supports range search with good pruning. The performance of the proposed indexing technique (Kd-tree with feature level fusion) has been compared with indexing based on Kd-tree with score level fusion. It is found that feature level fusion performs better than score level fusion and to get top matches for any query template.

Acknowledgment

The authors would like to thank Pradeep Nayak, Gaurav Gadhoke and Anuj Kumar Kaushal who have helped in collecting the data, and reviewers for their valuable comments and suggestions.

References

1. Jain, A.K., Flynn, P., Ross, A.: Handbook of Biometrics. Springer, Heidelberg (2008)
2. Mhatre, A., Chikkerur, S., Govindaraju, V.: Indexing Biometric Databases using Pyramid Technique. LNCS, pp. 841–849. Springer, Heidelberg (2005)
3. Mhatre, A., Palla, S., Chikkerur, S., Govindaraju, V.: Efficient search and retrieval in biometric databases. SPIE Defense and Security (2001)
4. Samet, H.: The Design and Analysis of Spatial Data Structures. Addison-Wesley, Reading (1990)
5. Guttman, A.: R-trees: A dynamic index structure for spatial searching. In: ACM SIGMOD Int. Conference on Management of Data, pp. 47–57 (1984)
6. Jolliffe, I.T.: Principal Component Analysis, 2nd edn. Springer, Heidelberg (2002)
7. Bentley, J.L.: Multidimensional binary search trees used for associative searching. Commun. ACM 18(9), 509–517 (1975)
8. Bentley, J.L.: K-d trees for semidynamic point sets. In: Proc. 6th Annual Symposium on Computational Geometry, pp. 187–197 (1990)
9. Bentley, J.L., Friedman, J.H.: Data structure for range searching. ACM Computing Surveys 11, 397–409 (1979)
10. Kaewkongka, T., Chamnongthai, K., Thipakorn, B.: Off-line Signature Recognition using Parameterized Hough Transform. In: Proc. of Int. Symposium on Signal Processing and Its Applications, vol. 1, pp. 451–454 (1999)
11. Jin, A.B., Ooi, O.T.S., Teoh, S.Y.: Offline Signature Verification through Biometric Strengthening.. In: IEEE Workshop on Automatic Identification Advanced Technologies, pp. 226–231 (2007)
12. Daugman, J.: How iris recognition works. IEEE Transactions on Circuits and Systems for Video Technology 14(1), 21–30 (2004)

13. Wildes, R.P.: Iris Recognition: An Emerging Biometric Technology. Proc. of the IEEE 85(9), 1348–1363 (1997)
14. Chen, T.-C., Chung, K.-L.: An Efficient Randomized Algorithm for Detecting Circles. Computer Vision and Image Understanding 83(2), 172–191 (2001)
15. He, X., Shi, P.: A novel iris segmentation method for hand-held capture device. LNCS, pp. 479–485. Springer, Heidelberg (2005)
16. Ma, L., Tan, T.N., Wang, Y.H., Zhang, D.: Local intensity variation analysis for iris recognition. Pattern Recognition 37(6), 1287–1298 (2004)
17. Feature Level Fusion Using Hand and Face Biometrics. In: SPIE conference on Biometric Technology for Human Identification II. vol. 5779 (2005)