Chapter 3
Efficient Score-Based Indexing Technique for Fast Palmprint Retrieval

Abstract Biometric identification systems capture biometric (i.e., fingerprint, palm, and iris) images and store them in a central database. During identification, the query biometric image is compared against all images in the central database. Typically, this exhaustive matching process (linear search) works very well for the small databases. However, biometric databases are usually huge and this process increases the response time of the identification system. To address this problem, we present an efficient technique that computes a fixed-length index code for each biometric image. Further, an index table is created based on the indices of all individuals. During identification, a set of candidate images which are similar to the query are retrieved from the index table based on the values of query index using voting scheme that takes less time. The technique has been tested on benchmark PolyU palmprint database and the results show a better performance in terms of response time and search speed compared to the state-of-the-art indexing methods.

Keywords Index code · Palmprint · SIFT · Sample images · Match scores

3.1 Introduction

This chapter presents a score-based indexing approach for the palmprints. The first such attempt was made by Maeda et al. [1]. They compute a match score vector for each image by comparing it against all the database images and stored these vectors permanently as a matrix. Though, the approach achieves quicker response time, it takes linear time in worst case and also storing of match score matrix leads to increase in the space complexity. Gyaourova et al. [2] improved the work on match scores by choosing a small set of reference images from the database. For every image in the database, a match score vector (index code) was computed by matching it against the sample set using a matcher and stored this match score vector as a row in an index table. However, a sequential search is done in the index space for identification of best matches which takes linear time and is prohibitive for a database containing millions of images. Paliwal et al. [3], used vector approximation (VA+) file to store the match score vectors and k-NN search, palmprint texture to retrieve best matches. However, the performance of VA+ file method generally degrades as dimensionality increases [4]. This chapter presents an indexing method that computes a fixed-length index code for each input biometric image based on match scores. The method proposes an efficient storing and searching method for the biometric database using these index codes. The proposed searching technique avoids the sequential scan on the database for identification and uses voting scheme, which results in a rapid search that takes less time.

3.2 Indexing

This section discusses our proposed methodology for indexing the biometric databases. The concept behind this approach is that if two palmprints p and q belong to same user, then their match scores (keys) against a third image (let s) are almost equal. This enables us to use these scores as index keys for the palmprints in an index table and arrange them like traditional records. To indentify a query palmprint, we compute its key (i.e., its match score with s) and retrieve the palmprint that have same key in the index table. However, many palmprints may have same key (i.e., match score against s) and mapped to the same bin of the index table. For example, alphabets X and Z are different but have same distance (score) to Y. This is shown in Table 3.1. Hence, a set of palmprints are retrieved as similar to the query palmprint. This retrieved set contains few palmprints that are not similar to q but have same score against s. Hence, multiple samples can be used to filter out these false matches. An overview of our proposed technique is shown in Fig. 3.1. The different steps involved in our approach are discussed in the following.

Table 3.1 Palmprints are arranged in ascending order of their scores against sample palmprint

Score (or Key)	List of Palmprints
0	$PList$
1	$PList$
2	$PList$
–	–
–	–
$x-1$	$PList$
x	$PList$
$x+1$	$PList$
–	–
–	–
100	$PList$

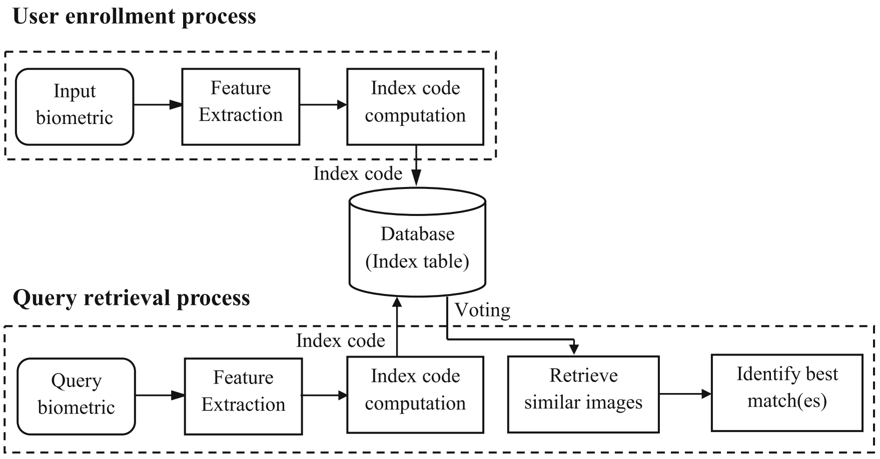

Fig. 3.1 Overview of the proposed approach

3.2.1 Feature Extraction

This section describes the key features used for the palmprint images. In this work, we use scale invariant feature transform (SIFT) points as the key features [5–8].

3.2.2 Index Code Computation

This section proposes an efficient method to compute indexes for biometric images that makes use of a sample image set. The index code computation process is shown in Fig. 3.2. An input image is compared against a set of sample images; the resultant set of match scores (i.e., keys) is called the index code (i.e., $INDEX$) of the input image [2].

This can be formulated as follows: Let a be an input image and $S = \{s_1, s_2, \ldots, s_k\}$ be the selected sample image set. Then, the index code of image a represented by $INDEX_a$ is given in Eq. 3.1, where $m(a, s_i)$ is the match score (i.e., key) of image a against i^{th} sample image.

$$\begin{aligned} INDEX_a &= \{m(a, s_1), m(a, s_2), \ldots, m(a, s_k)\} \\ &= \{key_1, key_2, \ldots, key_k\} \end{aligned} \tag{3.1}$$

The match score between two images is computed by comparing their key features in Euclidean space [9]. Note that $INDEX_a$ is the index code of image a and consists of k keys.

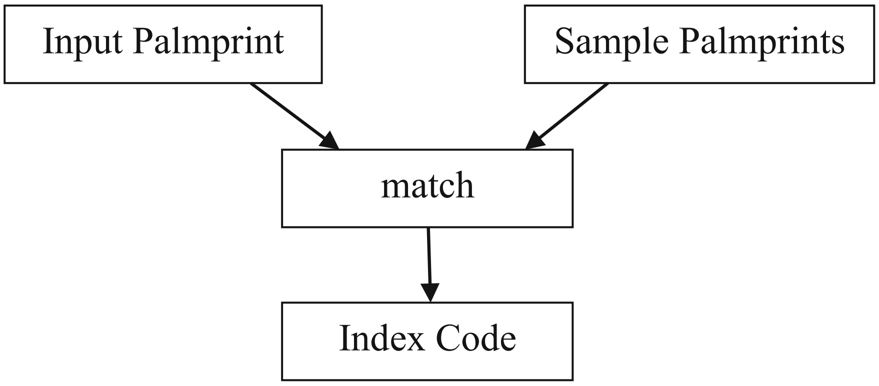

Fig. 3.2 Index code computation process

3.2.3 Index Table Creation and User Enrolment

To enroll the palmprints, a 2D Index Table A of size $100 \times k$ is created. Each column of the table corresponds to one sample image in the sample set. The match scores obtained are normalized in the range 0-100. For a given palmprint a, we compute the index code which consists of k keys. Let $x = m(a, s_i)$ be the key_i of palmprint a against sample image s_i. Then palmprint a is enrolled into bin $A(x, s_i)$. This process is repeated with other keys of palmprint a and is enrolled to the corresponding bin of the index table. The index table organization is shown in Table 3.2. It can be seen that each bin of the table $A(x, s_i)$ contains a list of palmprints (i.e., PList), whose match score is x against sample image s_i.

Table 3.2 Index table consists of $k + 1$ columns where the first column is the key and remaining k columns are corresponding to one palmprint of the sample set

Score (or Key)	s_1	s_2	–	s_k
0	$PList$	$PList$	–	$PList$
1	$PList$	$PList$	–	$PList$
2	$PList$	$PList$	–	$PList$
–	–	–	–	–
–	–	–	–	–
$x - 1$	$PList$	$PList$	–	$PList$
x	$PList$	$PList$	–	$PList$
$x + 1$	$PList$	$PList$	–	$PList$
–	–	–	–	–
–	–	–	–	–
100	$PList$	$PList$	–	$PList$

We illustrate the palmprint enrollment process with an example. Let $S = \{s_1, s_2, s_3\}$ be a sample set of three palmprints (i.e., $k = 3$). Thus, the index code of a palmprint consists of three keys each against one sample image. Hence, the index table requires three columns. Let a be an input palmprint and {45, 59, 35} be its index code. We use the first key, i.e., 45 and enroll a into (45, 1) location in the index table A. Then, using the second key 59, we access $A(59, 2)$ bin and enroll the palmprint identity a into the $PList$. Finally, using third key 35, we enroll a into $PList$ at $A(35, 3)$ of index table. The other palmprints of the database are also enrolled into the index table likewise. Finally, each column of the index table contains all palmprints in the order of their keys against corresponding sample image (Table 3.2). Algorithm 3.1 explains the process of enrolling a palmprint into index table.

Algorithm 3.1 Palmprint enrollment process

```
1: INPUT: Input Palmprint $a$, Sample image set $S = \{s_1, s_2, .., s_k\}$, Index Table $A(100 \times k)$.
2: OUTPUT: Updated $A$
   // Enroll Palmprint $a$ into $A$
3: for each $s_i \in S$ do
4:     $x \leftarrow m(a, s_i)$            // $m(a, s_i)$ is the match score (i.e., key) of $a$ against $s_i$
5:     $A(x, i).PList \leftarrow a$        //Enroll $a$ into $PList$ at location $(x, i)$ of $A$.
6: end for
7: RETURN Updated $A$.
```

3.3 Retrieval of Best Matches for a Query

This section proposes an efficient retrieval system to identify a query image from the index table. During identification, the technique retrieves a set of palmprint identities (i.e., candidate list) from the index table which are most similar to the query using voting method. To do this, we first compute the index code of the query. The index code of query image q represented as $INDEX_q$ is given in Eq. 3.2.

$$INDEX_q = \{m(q, s_1), m(q, s_2), \ldots, m(q, s_i), \ldots, m(q, s_k)\} \tag{3.2}$$

Let $x = m(q, s_i)$ be the i^{th} match score value of the query index code, the algorithm uses x as key to the index table and retrieves all the palmprints $(PList)$ found in the bin $A(x, i)$. We also retrieve palmprints from the predefined neighborhood λ of the selected bin in the corresponding column i to handle the natural distortions. Finally, we give a vote to each retrieved image. We repeat this process with other keys of the query index code. In our next step, we accumulate and count the number of votes of each palmprint identity. Finally, we sort all the individuals in descending order based on the number of votes received. We select the individuals whose vote score is greater than a predefined threshold as best matches (candidate list) to the query image.

The query identification process can be illustrated using an example. Let q be a query palmprint and $INDEX_q$ = {35, 54, 56}. We use key_1 which is 35 to access 35th bin in first column of index table (i.e., $A(35, 1)$) and retrieve all the palmprints found there. Next we also retrieve palmprints from a predefined neighborhood λ. Let $\lambda = 2$. Hence, the range of locations is from 33 ($35 - \lambda$) to 37 ($35 + \lambda$) in column 1. We add all these palmprints from locations 33 to 37 into temporary list L. Then, using key_2, we access $A(54, 2)$ and retrieve the palmprints from bins 52 to 56 into L. Further, using key_3, we access $A(56, 3)$ and retrieve the palmprints from bins 54 to 58 into L. Finally, we select the palmprints that have appeared more number of times (receives more votes) than a predefined threshold as potential candidates to q.

The motivation to this voting mechanism is that, if two palmprints (say query and an enrolled palmprint) belong to the same hand, then their scores against a sample palmprint are similar. So the assumption is that, if the database contains the query palmprint identity, the total number of votes received for this is more than other enrolled palmprints.

Algorithm 3.2 Palmprint identification: Retrieving the best match for a query

```
1: INPUT: Query Palmprint $q$, Sample image set $S = \{s_1, s_2, .., s_k\}$, Index Table $A(100 \times k)$,
   Predefined neighborhood $\lambda$.
2: OUTPUT: Candidate Set $C$
   //Retrieve set of similar palmprints to $q$ from $A$
3: $L \leftarrow \{\}$
4: for each $s_i \in S$ do
5:   $x \leftarrow m(q, s_i)$                       // $m(q, s_i)$ is the key of $q$ against $s_i$
6:   for $j = x - \lambda$ to $x + \lambda$ do
7:     $L \leftarrow L \cup A(j, i).PList$          // $L$ is a temporary list.
8:   end for
9: end for
10: Retrieve the $P_{ids}$ in $L$ whose Vote score greater than a predefined threshold $T$ as similar to $q$
    and retrieve them into $C$.
11: RETURN Candidate Set $C$.
```

3.4 Selection of Sample Images

The selection of sample images from the database plays a crucial role in the performance of the system. Images which are more different from one another and represent the qualities of the entire database should be selected as sample images. The sample images should be selected such that they are having distinct characteristics and provide enough information about the database for identification with minimal computational cost [3]. In this work, we explore two different methods for selection of representative images: (a) Max-variance method, (b) k-means clustering.

Let D be the database corresponding to n users. We divided the database into two datasets: $Gallery$ (i.e., Training) set consists of M images and $Probe$

(i.e., Testing) set consists of N images. For each image in $Gallery$, determine its variance of grayscale intensity values. Let $Gallery = \{U_1, U_2, \ldots, U_n\}$, where U_i is a set of images of subject i. From each set U_i, select one image which is having maximum variance in the set (i.e., the image of better quality). Let $A = \{a_1, a_2, \ldots, a_n\}$ be the set of all selected images. The set A contains the candidates from which sample image set $S = \{s_1, s_2, \ldots, s_k\}$ is selected where $S \subset A$, and $|S| \ll |A|$.

3.4.1 Max-variance Method

The max-variance method sorts the images in set A in descending order based on the variance and selects the top k images (k is determined empirically) for the representative image set S. The idea behind choosing such representative set is that the highly variant images contain significant properties that represent the various qualities of the database.

3.4.2 k-Means Clustering

The second method relies on the concept of clustering. This method partitions the set X into k clusters based on variance using k-means clustering such that images in the same cluster are similar to each other; whereas images in different clusters are dissimilar. As each cluster contains similar images, one image from each cluster which is closer to the cluster centroid is selected for set S as sample of that cluster. Unlike the max-variance method which selects sample images from a single group (i.e., the images with maximum variances), the clustering method selects images from different groups and thus satisfies the aforementioned property.

3.5 Experimental Results

This section shows the performance of the proposed indexing approach experimentally. The experiments are conducted on PolyU palmprint database [10].

First, we validate different parameters such as neighborhood size (λ), selection rules for the sample palmprints, etc., that are involved in this work.

3.5.1 Neighborhood Size (λ)

The neighborhood size λ plays a major role on the system performance. An experiment is conducted by varying the λ values from 0 to 8 and observed the system miss

Table 3.3 Effect of neighborhood size λ on indexing performance

λ	MR	PR
0	55.63	12.42
1	52.46	14.56
2	48.53	16.19
3	38.62	18.55
4	33.06	20.62
5	**30.03**	**26.82**
6	27.52	33.91
7	20.13	35.77
8	12.28	38.06

rate (MR) and penetration rate (PR). This is shown in Table 3.3. It is observed that, the PR increases with λ while MR decreases. Hence, the optimum value for the λ is chosen as a point where the MR and PR values are approximately equal which is 5.

3.5.2 Selection Rules for Sample Palmprints

The sample images should be very different from another and represent entire qualities of the database. Hence, an experiment is conducted to validate the system performance using various rules for the selection of sample images. Four different selection rules are considered (Fig. 3.3):

1. Max-variance approach
2. k-means approach
3. Randomly selected k palmprints
4. First k palmprints of the database

The proposed max-variance and k-means algorithms achieve less PR and high HR compared to other approaches. This performance (Fig. 3.3) shows the superiority of proposed rules for the selection of sample palmprints. Further, it can be observed that the proposed k-means clustering rule performs better than the max-variance method. This shows the ability of the k-means clustering approach for retrieving the sample palmprints from the database.

3.5.3 Results and Performance Comparison

This section describes the results of the proposed approach and its comparison with the prominent approaches in the literature. The HR and PR of the system at various thresholds (T = 1,2,..,100) were determined and shown in Fig. 3.3. It can be seen that,

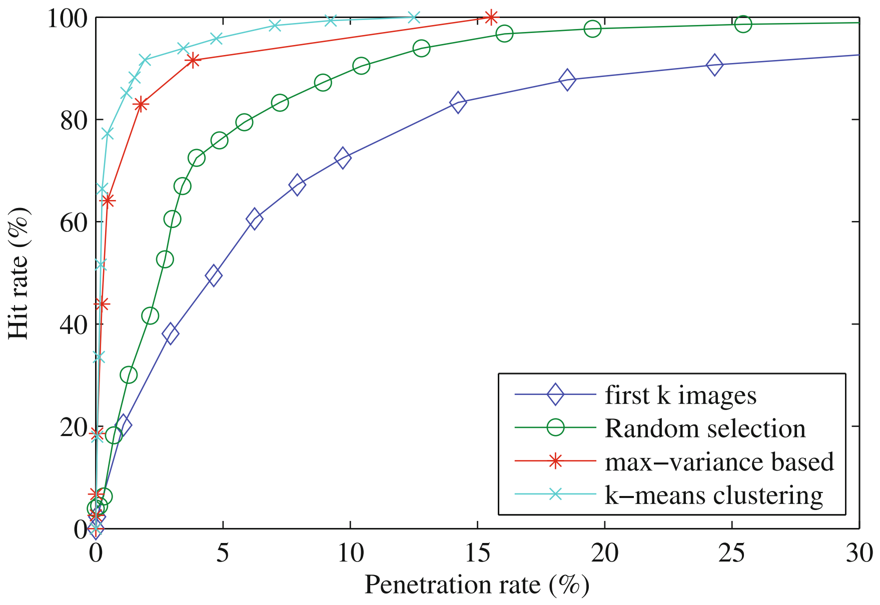

Fig. 3.3 Performance of the system using different selection rules for representative images over PolyU database

at HR=100%, the PR of our method is 12.5 and 15.54% for the *k*-means approach and max-variance approaches, respectively. In other words, our retrieval algorithms searches only 12.5 and 15.54%, of the database and the genuine image is identified with a probability (i.e., HR) of 100%.

Further, the proposed approaches are compared with Paliwal et al. [3] method and Badrinath et al. [11] method. Paliwal et al. [3] approach is also a match score based method. They used the VA+ file method to store the index codes. This approach chose 171 palmprints for the sample set and achieved an HR of 98.28% only. On the other hand, proposed methods used 130 sample palmprints and achieved 100% HR. Badrinath et al. [11] used SURF features from the palmprints and indexed them using geometric hashing [12]. But, they achieved a PR of 31.89% only [11]. Table 3.4 shows the performance of various approaches.

Table 3.4 PR (%) of the system at maximum HR (%) achieved using different techniques

Approach	HR	PR
Badrinath et al. [11]	100	31.89
Paliwal et al. [3]	98.28	–
Proposed *k*-means	100	12.5
Proposed Max-variance	100	15.54

3.5.4 Retrieval Time

The retrieval time of the proposed system is analyzed using big-O notation. As shown in Algorithm 3.2, to identify the potential candidates C for a query palmprint q, it is matched against each sample palmprint s_i and it retrieves the $PList$ from the mapped bin and its neighborhood to temporary list L. Note that, this process requires O(k) time as there are k sample palmprints. Note $k \ll N$ where N is the size of the database. In the next step, the P_{id}s that are repeated more times in L are retrieved into candidate set C. Let m be the size of L. This process requires O(m) time. Note that $m \ll N$.

Therefore, the retrieval time of this approach can be approximated as $O(k)+O(m)$ time. On the other hand, a linear search method requires O(N). Thus, we conclude that the proposed algorithm takes less time for retrieval of candidate set than linear search method because $(k + m) < N$.

3.6 Summary

In this chapter, an efficient indexing algorithm for palmprint databases using fixed-length index codes is proposed. We propose an efficient storing method for the biometric database using these index codes such that they are sorted like traditional records and retrieved the best matches similar to the query in a less time. Two different selection approaches are used for choosing the sample palmprints and showed their effectiveness on the performance. The proposed system avoids the sequential scan and use voting to retrieve the best matches. Further, without compromising identification performance, our algorithm performs well than prominent indexing methods. This approach is easy to implement and can be applied to any biometric database.

References

1. T. Maeda, M. Matsushita, and K. Sasakawa. **Characteristics of the Identification Algorithm Using a Matching Score Matrix**. In *ICBA*, pages 330–336, 2004.
2. A. Gyaourova and A. Ross. **Index Codes for Multibiometric Pattern Retrieval**. *IEEE Transactions on Information Forensics and Security*, **7**(2):518–529, 2012.
3. A. Paliwal, U. Jayaraman, and P. Gupta. **A score based indexing scheme for palmprint databases**. In *International Conference on Image Processing*, pages 2377–2380, 2010.
4. R. Weber, H.J. Schek, and S. Blott. **A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces**. In *VLDB*, **98**, pages 194–205, 1998.
5. SIFT. **SIFT for matlab:**. http://www.vlfeat.org/vedaldi/code/sift.html.
6. D.G. Lowe. **Distinctive Image Features from Scale-Invariant Keypoints**. *International Journal of Computer Vision*, **60**(2):91–110, 2004.
7. G.S. Badrinath and P. Gupta. **Palmprint Verification using SIFT features**. In *First Workshop on Image Processing Theory, Tools and Applications*, pages 1–8, 2008.

8. Q. Zhao, W. Bu, and X. Wu. **Sift-based image alignment for contactless palmprint verification**. In *2013 International Conference on Biometrics*, pages 1–6, 2013.
9. Ilaiah Kavati, Munaga VNK Prasad, and Chakravarthy Bhagvati. **A Score-Based Indexing and Retrieval Technique for Biometric Databases**. *International Journal of Pattern Recognition and Artificial Intelligence*, page 1756009, 2016.
10. PolyU. **The PolyU palmprint database:**. http://www.comp.polyu.edu.hk/biometrics
11. G.S. Badrinath, P. Gupta, and H. Mehrotra. **Score level fusion of voting strategy of geometric hashing and SURF for an efficient palmprint-based identification**. *Journal of real-time image processing*, **8**(3):265–284, 2013.
12. H.J. Wolfson and I. Rigoutsos. **Geometric Hashing: An Overview**. *IEEE Comput. Sci. Eng.*, **4**(4):10–21, 1997.