Palmprint Indexing Based on Ridge Features

Xiao Yang, Jianjiang Feng, Jie Zhou
Department of Automation
Tsinghua University, Beijing, China
xiao-yang09@mails.tsinghua.edu.cn, jfeng@tsinghua.edu.cn, jzhou@tsinghua.edu.cn

Abstract

In recent years, law enforcement agencies are increasingly using palmprint to identify criminals. For law enforcement palmprint identification systems, efficiency is a very important but challenging problem because of large database size and poor image quality. Existing palmprint identification systems are not sufficiently fast for practical applications. To solve this problem, a novel palmprint indexing algorithm based on ridge features is proposed in this paper. A palmprint is pre-aligned by registering its orientation field with respect to a set of reference orientation fields, which are obtained by clustering training palmprint orientation fields. Indexing is based on comparing ridge orientation fields and ridge density maps, which is much faster than minutiae matching. Proposed algorithm achieved an error rate of 1% at a penetration rate of 2.25% on a palmprint database consisting of 13,416 palmprints. Searching a query palmprint over the whole database takes only 0.22 seconds.

1. Introduction

Palmar skin of human is covered with two unique patterns, namely, the palmar friction ridges and the palmar flexion creases [9]. These two types of patterns are claimed to be permanent and unique to an individual [2], indicating the value of palmprints for personal identification. Palmprints have a much larger valid area and contain much more minutiae than fingerprints, indicating that palmprints are more distinctive than fingerprints [11]. In addition, more than 30% of the prints obtained from crime scenes are from palms [7]. Developing a national palmprint identification system has become a main objective of the FBI's Next Generation Identification (NGI) program [1].

Early palmprint recognition systems for civilian applications were developed mostly based on low-resolution (about 100 ppi) images [17, 20]. These palmprints are generally captured with contactless devices and some of them use pegs to fix the position of hands for pre-aligning different palms. In these systems recognition is mainly based on comparing crease features.

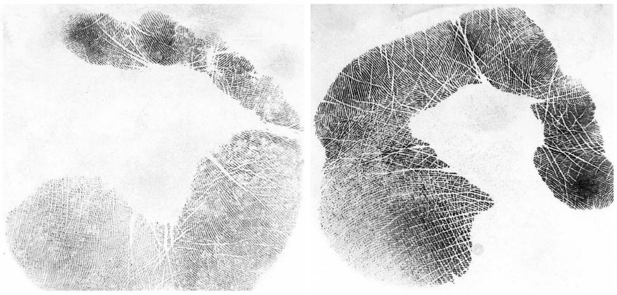


Figure 1. Typical palmprints in a forensic palmprint database: the palm is not properly aligned, central area of the palm is often missing, and image quality is poor.

Ridge feature based palmprint matching is required in law enforcement applications because these recognition systems need to identify poor quality partial palmprints left at crime scenes as well as meet the requirements of courts of law. However, due to complexity in sensing, storage and recognizing palmprints, ridge feature based palmprint matching systems have lagged behind of Automated Fingerprint Identification Systems (AFIS) [16], which are also based on ridge features, mainly minutiae. In recent years, with the technical development of live-scan devices and computer hardware, high-resolution (500 ppi) palmprints are being collected by more and more law enforcement agencies. Accordingly public research on ridge feature based palmprint matching began to emerge. Jain and Feng [9] proposed the first ridge feature based palmprint matching algorithm, which is composed of a crease-insensitive ridge orientation field estimation algorithm and a descriptor based minutiae matching algorithm. Dai and Zhou improved the matching algorithm in [6] by combining minutiae features with additional ridge features and crease features.

Although these palmprint matching systems [6, 9] have achieved good matching accuracy, they are not sufficiently efficient for identifying palmprints in a large database. The average time of matching a pair of full palmprints is about

978-1-4577-1359-0/11/$26.00 ©2011 IEEE

2 seconds in [9] and 5.1 seconds in [6]. Suppose that we want to search a query palm in a dataset which contains 1 million palms, it will take 11 days and 28 days respectively for [9] and [6]. Considering the huge size of law enforcement palmprint databases and the large number of query requests, the efficiency of existing palmprint matching systems needs to be improved significantly.

A solution to this problem is palmprint classification (classify palmprints into disjoint categories) or indexing (or continuous classification [14]). Unlike fingerprints, however, there is no widely accepted classification schemes for palmprints. Existing research on palmprint classification is mostly focused on low-resolution images. Wu *et al.* [18] classify palmprints into six categories according to the number of the principal lines (or major creases) and the number of their intersections. Nevertheless, the six categories are not evenly distributed and 78% of all the palms in their dataset belong to Category 5. This is unlikely to improve the matching efficiency significantly. Fang *et al.* improved the algorithm in [8] by classifying Category 5 into five sub-categories depending on where the intersection point falls in. Unfortunately, these methods cannot be used for palmprints in law enforcement applications because of poor image quality and incompleteness of many palmprints, as shown in Figure 1.

Another related topic is fingerprint classification and indexing [15]. Fingerprints can be classified into five categories, plain arch, tented arch, left loop, right loop, and whorl, based on distribution of singular points (loops and deltas). However, singular point based classification scheme is not suitable for palmprint classification because most singular points in palmprints are located on the border and thus are not available in many images. So called continuous fingerprint classification [14] represents the orientation field and/or the density map of a fingerprint as a feature vector by defining a common coordinate system using the core point in fingerprints. However, there is no stable reference points in palmprints.

In this paper, we propose a fast palmprint indexing system, which is based on palmprint pre-alignment and coarse matching. Palmprint pre-alignment refers to aligning palmprints in the common coordinate system in the feature extraction stage. This constitutes the first speedup strategy since matching pre-aligned palmprints is much faster than matching original palmprints. In coarse matching, the system finds a list of candidate palmprints by comparing ridge orientation field and density/frequency map, which is much faster than matching minutiae.

The proposed indexing system is tested on a dataset with 13,416 full palmprints. At a penetration rate of 2.25%, the error rate of the proposed approach is only 1%. Note that the accuracy of the state of the art fingerprint indexing algorithms is 1% error rate at a penetration rate of 30% [4]. It takes only 0.22 seconds for the proposed indexing system to search one query palmprint in a gallery set of 12,716 palmprints.

2. Palmprint Indexing

The idea of the proposed palmprint indexing system is illustrated in Figure 2. In the enrollment stage, a gallery palmprint is aligned with respect to a set of k reference orientation fields, which are found by a clustering algorithm in the offline stage, and k indices (pre-aligned orientation field and density map) of this palmprint are generated. In the retrieval stage, k indices of a query palm are generated by the same procedure. To estimate the similarity between a query and a gallery palmprint, k similarities are calculated between k pairs of corresponding indices and the maximum one is selected as the similarity. The top M most similar gallery palmprints are returned as the candidates of the query palmprint.

This section discusses the indexing phase while the retrieval phase is covered in the next section. In the following subsections, we first introduce the feature extraction algorithm and the notation briefly. Then reference orientation field based palmprint pre-alignment is described. Finally we detail multi-reference based palmprint pre-alignment.

2.1. Feature Extraction

As we state below, the orientation field is used for palmprint pre-alignment while both orientation field and density map are used as index. We use the method in [9] to extract these features from palmprints.

The feature extraction algorithm [9] partitions each palmprint into $n_B = n_H \times n_W$ blocks and outputs blockwise orientation field O, density map D and quality map Q. Let $o(m)$, $d(m)$ and $q(m)$ represent the ridge orientation, ridge period (the inverse of ridge frequency) and quality in the block located at (h, w) (where $m = h \times n_W + w$) in the blockwise palmprint respectively. Q is a binary image with 0 indicating background.

2.2. Reference Based Palmprint Pre-alignment

In existing fingerprint retrieval systems (*e.g.* [12]), a fingerprint is pre-aligned with respect to a stable reference point. Then, similarity between fingerprints are directly calculated by comparing the pre-aligned orientation field and density map. Thus the indexing performance heavily depends on the pre-alignment. Unfortunately, it is very hard to find a universal definition of reference point for palmprints. In addition, palmprints may lose some area when it is captured. Some researchers use inter-finger gaps [20] or the endpoints of major creases [21] to pre-align palmprints. However, these feature points are not stable in forensic palmprints.

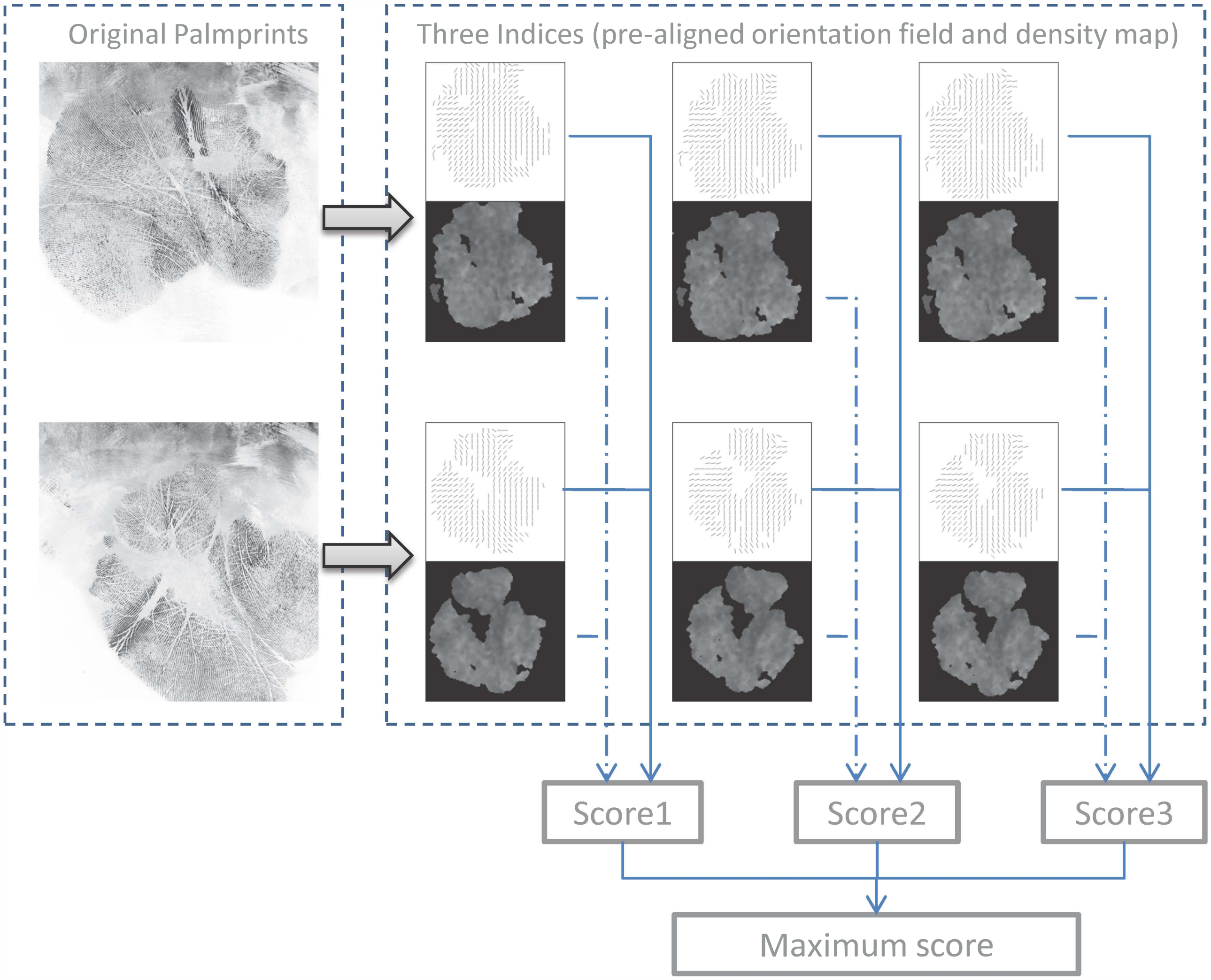

Original Palmprints
Three Indices (pre-aligned orientation field and density map)
Score1
Score2
Score3
Maximum score

Figure 2. Proposed palmprint indexing system

Figure 3 shows the similarity of orientation fields of four randomly selected palmprints, from which we can find that orientation fields of different palms are similar in many positions of the palmar region. This similarity suggests that different palmprints can be registered in the same coordinate system by aligning their orientation fields with respect to a common orientation field, termed as reference orientation field.

Yager and Amin [19] use the Generalized Hough Transform algorithm (GHT) [3] to estimate the spatial transformation between the orientation fields of a query fingerprint and a template fingerprint. We further extend this idea by 1) applying GHT to palmprint orientation field and 2) prealigning any palmprint by aligning its orientation field to a common reference orientation field.

The GHT-based orientation field alignment is described below. The Hough Transform vote space is defined as $f(\Delta x, \Delta y, \varphi)$, in which Δx,Δy and φ are restricted by predetermined ranges respectively. The value $f(\Delta x, \Delta y, \varphi)$ indicates the number of possible matched element pairs with the parameter $(\Delta x, \Delta y, \varphi)$. First, f is initialized as 0. Then for each pair of potentially matching elements between the two orientation fields, the transformation parameter is calculated and the corresponding bin in the Hough space receives a vote. Finally, the bin with the maximum value is chosen as the transformation parameter between two orientation fields.

2.3. Multi-Reference Based Palmprint Prealignment

The goal of palmprint pre-alignment is to consistently align the different prints of the same palm. It is difficult to

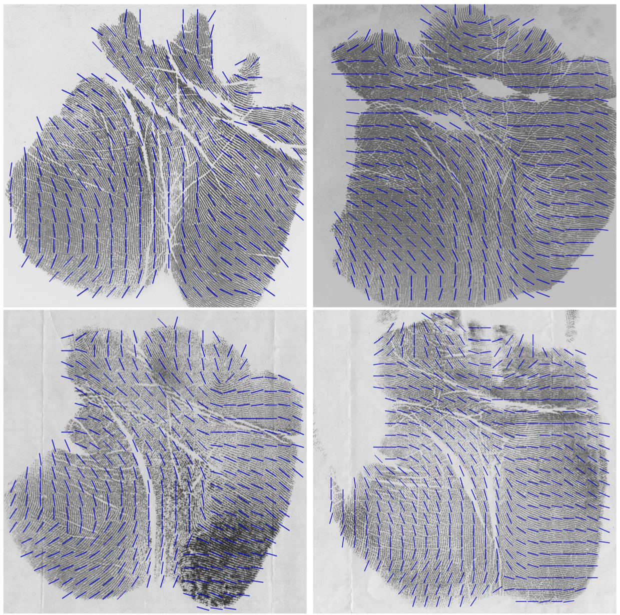


Figure 3. Similarity of different palmprint orientation fields.

achieve this goal using a single reference orientation field (as is experimentally shown in section 4.2.2). Thus we find k references using a clustering algorithm and utilize multi-reference method to pre-align palmprints more consistently.

The selection of reference orientation fields is very important for pre-alignment. We wish to use a few reference orientation fields to represent various orientation fields. Out of this consideration, we find reference orientation fields through clustering a set of given orientation fields.

The k-centroids clustering algorithm (also called k-medoids clustering) [13] is used in this work. Compared with the k-means clustering, the cluster center of k-centroids is a real palmprint orientation field instead of an averaged artificial one. Given the orientation fields of a set of N_T training orientation fields, k ($k < N_T$) reference palmprints are found by k-centroids clustering algorithm as follows:

1. k initial cluster centers ($k/2$ right palms and $k/2$ left palms) $O_j^C, j = 1, ..., k$ are randomly selected and each one represents a cluster.

2. Each training orientation field $O_i^T, i = 1, ..., N_T$ is compared to each cluster center $O_j^C, j = 1, ..., k$ to obtain k similarity scores, $S_O\left(O_i^T, O_j^C\right)$ (similarity measure of orientation field is described in section 3). Then O_i^T is assigned to the cluster j^* if $S_O\left(O_i^T, O_{j^*}^C\right)$ is the maximum in $S_O\left(O_i^T, O_j^C\right), j = 1, ..., k$.

3. In each cluster j, re-select the center orientation field by 1) setting each non-center orientation field O^{temp} as the center in turn, 2) calculating the total similarity score between all orientation fields in cluster j and O^{temp}, and 3) choosing the orientation field with the highest similarity score as the new center of cluster j.

4. Repeat step 3 on all k clusters and k new center orientation fields are selected. If all k centers are the same as the last step, the clustering process is finished and k center/reference orientation fields are returned, otherwise go to step 2.

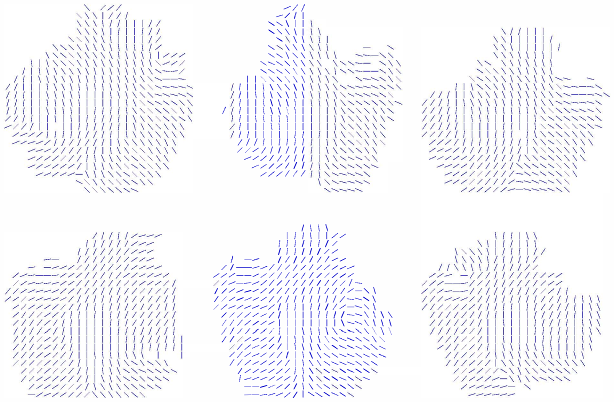


Figure 4. Six reference orientation fields found by the clustering algorithm. The orientation fields on the first row are generated from right palms, while those on the second row are from left palms.

Six reference palmprint orientation fields found by the above clustering algorithm are shown in Figure 4.

With the k reference orientation fields $O_j^R, j = 1, ..., k$, we can generate index for any given palmprint. We first estimate the transformation parameters between the given palmprint and each reference using the GHT algorithm. Then we align the orientation field and density map of the given palmprint using each of the k estimated transformations. The k versions of orientation field and density map are directly used as the indices of the given palmprint.

3. Palmprint Retrieval

The similarity of two orientation fields O_1 and O_2 is denoted as $S_O(O_1, O_2)$. It is used in clustering and retrieval and is computed by

$$S_O\left(O_1, O_2\right) = \frac{\sqrt{\left(\sum_m s_{12}(m)\cos\delta_{12}(m)\right)^2 + \left(\sum_m s_{12}(m)\sin\delta_{12}(m)\right)^2}}{n_B/5 + \sum_m s_{12}(m)} \quad (1)$$

where $s_{12}(m) = q_1(m) \times q_2(m)$ and $\delta_{12}(m) = 2(o_1(m) - o_2(m))$. We can easily find that $s_{12}(m) = 1$ indicates the m^{th} ($m = 1, ..., n_B$) block of two palmprints are both available and $\sum_m s_{12}(m)$ is the total number of pairwise available blocks. As mentioned above, palmprints may lose some area, which means the number of available blocks should be taken into account. Hence, S_O is set as 0 if the number of available blocks is lower than 10% of the total block number n_B, and $n_B/5$ is added in the denominator to penalize the cases of small number of available blocks.

The similarity based on density map $S_D(D_1, D_2)$ is computed by

$$S_D\left(D_1, D_2\right) = 1 - \frac{\sum_m s_{12}(m)\frac{|d_1(m)-d_2(m)|}{\max(d_1(m),d_2(m))}}{\sum_m s_{12}(m)} \qquad (2)$$

Also, the score is set 0 if the number of available blocks is lower than 10% of the total block number n_B.

Finally, we fuse the two scores into a final match score using the weighted sum rule:

$$S = \omega S_O\left(O_1, O_2\right) + (1-\omega)S_D\left(D_1, D_2\right) \qquad (3)$$

where $0 \leq \omega \leq 1$ and it is determined as 0.08 using a training set (see section 4.2.1).

To compute the similarity between two palmprints, we obtain k match scores between k pairs of pre-aligned orientation fields and density maps, and choose the maximum score as the final match score.

4. Experimental Results

In the following subsections, we first describe the database and performance indicators, and then report experimental results on parameter selection, indexing accuracy and efficiency.

4.1. Database and Performance Indicators

A large palmprint database is collected to test the proposed algorithm. The database is composed of two parts: 166×8 prints collected from 166 different palms in our lab (referred to as multi-impression dataset) and 13,616 different palmprints collected by a law enforcement agency (referred to as single-impression dataset). All images are captured at 500 ppi.

To evaluate the performance of proposed indexing algorithm, we choose the error rate vs penetration rate curve and the average penetration rate as evaluation criterions. These two criterions have also been used to evaluate in fingerprint indexing approaches (*e.g.* in [4, 12]). The penetration rate is defined as the average percentage of the database that the fine matcher (*e.g.* minutiae matcher) has to search. The error rate indicates the percentage of the query palmprints whose mated galley palmprints are not found at a certain penetration rate. The average penetration rate is an indicator in incremental search [5] in which the search stops as soon as the true candidate is found. There is no retrieval error in incremental search and the average penetration rate is defined as the mean of the penetration rates of all queries.

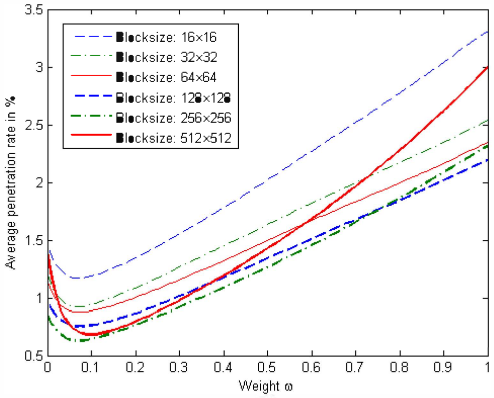


Figure 5. Average penetration rate for different weights and block sizes.

4.2. Parameter Selection

To select proper parameters (block size, weight ω for score fusion, and number k of reference orientation fields), we use 66×8 prints from the multi-impression dataset and 1000 prints from the single-impression dataset as the training dataset.

4.2.1 Block Size and Weight for Similarity Fusion

The block size of orientation field and density map influences both the accuracy and efficiency of the proposed indexing algorithm, while the weight in Equation 3 impacts the indexing accuracy. To determine the optimal combination of these two parameters, we evaluate the average penetration rate of the proposed indexing algorithm with different parameters. A set of 8 indexing experiments is conducted on the training set. In the i^{th} experiment, the i^{th} print of the 66 multi-impression palms is assumed to be the gallery print while the other 7 prints are assumed to be the query prints, and the 1000 prints from the single-impression dataset are used as the background.

We test the proposed algorithm with different ω from 0 to 1 with a step 0.01 and six different block sizes, which are 16×16, 32×32, 64×64, 128×128, 256×256, and 512×512 pixels. The low resolution orientation fields and density maps are down-sampled versions of the original orientation field and density map at the block size of 16×16 pixels. The average penetration rates for different ω and different block sizes are shown in Figure 5. The result shows that the lowest average penetration rate is achieved by blocks of 256×256 pixels. The lowest average penetration rates for different block sizes and corresponding weights are shown in Table 1, from which we can see proper down-sampling can reduce the impact of noise and make the algorithm more robust. On the other hand, down-sampling significantly re-

Table 1. Lowest average penetration rate and corresponding weight for different block sizes.

Block size in pixels	Lowest average penetration rate	Corresponding weight ω
16×16	1.17%	0.07
32×32	0.93%	0.07
64×64	0.88%	0.07
128×128	0.76%	0.07
256×256	0.63%	0.08
512×512	0.69%	0.10

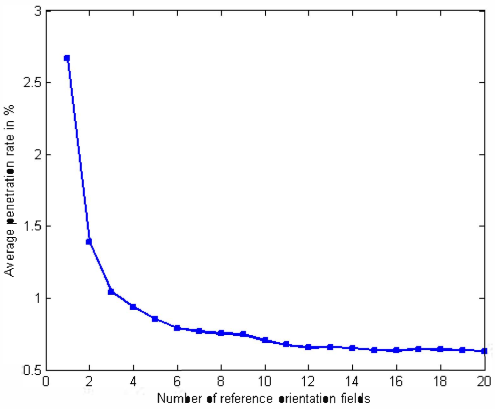


Figure 6. Average penetration rate for different numbers of reference orientation fields.

duces the dimensionality of the feature vectors and speeds up the indexing. Note that $\omega = 0$ means that only the score S_D is used, while $\omega = 1$ means that only the score S_O is used. At the block size of 256×256, if we use only S_D or S_O, the average penetration rate is 0.86% and 2.32% respectively. We select $\omega = 0.08$ and the average penetration rates is 0.63%, reduced by 26.74% and 72.84% compared with the result of using S_D or S_O alone respectively. From this experiment (and also Figure 8), we observed an interesting difference between fingerprint and palmprint, namely, fingerprint orientation field produces higher indexing accuracy than fingerprint density map [4], while it is reverse in palmprint indexing.

4.2.2 Number of References

The number, k, of reference orientation fields influence both the accuracy and efficiency of our algorithm. While a large k is beneficial for the indexing accuracy, a small k is desired for short response time. To choose k, we evaluate average penetration rate for different k using the training set.

Figure 6 illustrates the tradeoff between accuracy and efficiency. From this figure, we observed that for small k, the error rate drops significantly with the increase of k; But when $k > 5$, the error rate curve becomes almost flat. Thus, we choose $k = 20$ in our experiments.

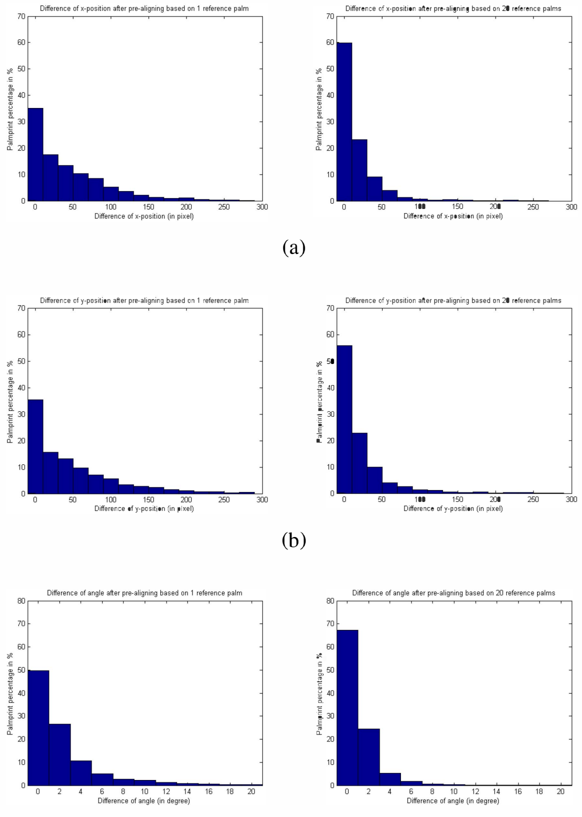
The better indexing accuracy of larger k is due to higher pre-alignment consistence among different prints of the same palm. To verify this hypothesis, another experiment is conducted on the same 66×8 prints from the multi-impression dataset. The consistence of pre-alignment is measured by comparing the relative transformation between each pair of prints from the same estimated by the proposed multi-reference algorithm with the ground truth transformation estimated by a classic GHT based minutiae matching algorithm [10].[1] The histograms of the difference in translation and rotation are separately plotted for two versions ($k = 1$ and $k = 20$) of the proposed pre-alignment algorithm. From Figure 7 we can see that the multi-reference method observably improves the consistence of pre-alignment.

4.3. Indexing Accuracy

To test the indexing accuracy of proposed algorithm, experiment is conducted on an independent dataset, in which the other 100×8 palmprints from the multi-impression dataset and 12,616 prints from the single-impression dataset are used. A set of 8 experiments is conducted. In the i^{th} experiment, the i^{th} print of the 100 multi-impression palms is assumed to be the gallery print while the other 7 prints are assumed to be the query prints, and the 12,616 prints from the single-impression dataset are used as the background. The parameters used in this experiment are: $\omega = 0.08$, block size $= 256 \times 256$ pixels, and $k = 20$.

The indexing performance is evaluated by the error rate vs penetration rate curve and shown in Figure 8. The three curves in Figure 8 correspond to the performance of three different features: orientation field, density map, and their fusion. The error rate reaches 1% at the penetration rate of 2.25%, while the state of the art fingerprint indexing algorithm [4] reaches the same error rate at a penetration rate of 30%. That means the proposed palmprint indexing system is 10 times more accurate than the state of the art fingerprint indexing system. The average penetration rate of the proposed algorithm is 0.113%, while the state of the art fingerprint indexing algorithm reported the best result of 0.94% [4]. Figure 9 shows an example of successful retrieval result. In this figure, palmprint on the left is a query while the one on the right is its corresponding gallery print, which ranks 1st in the whole database. Figure 10 shows an unsuccessful example, where left palmprint is query print

[1] The transformation estimated by a minutiae matching algorithm can serve as ground truth because (i) the transformation is pairwisely estimated between each pair of prints of the same palm rather than separately estimated between each print and a reference which are not from the same palm and (ii) precise transformation can be estimated based on matched minutiae.



(a)

(b)

(c)

Figure 7. Consistence of pre-alignment for different numbers of references. Left: $k = 1$; Right: $k = 20$. (a) Difference in horizontal translation, (b) difference in vertical translation, (c) difference in rotation.

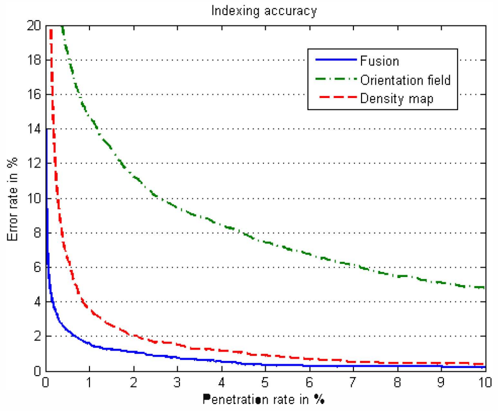


Figure 8. Indexing accuracy of the proposed method on the database consisting of 12,716 palmprints.

and the right one is its corresponding gallery print, which requires a penetration rate of 40.92% in the whole database. The unsuccessful retrievals are mainly due to inconsistent pre-alignment between the query palmprints and the corresponding gallery palmprints. If the common area between the query and corresponding gallery palmprint is small and not clear (as Figure 10 shows), the pre-alignment tends to be inconsistent.

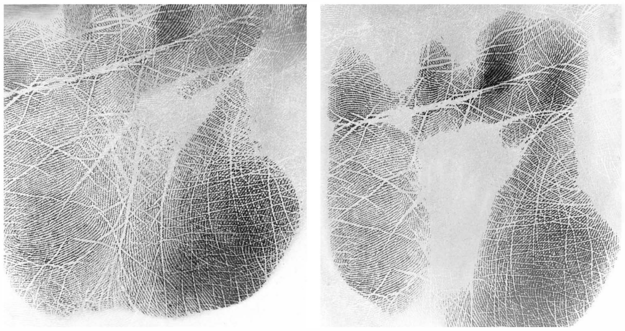


Figure 9. An example of successful retrieval

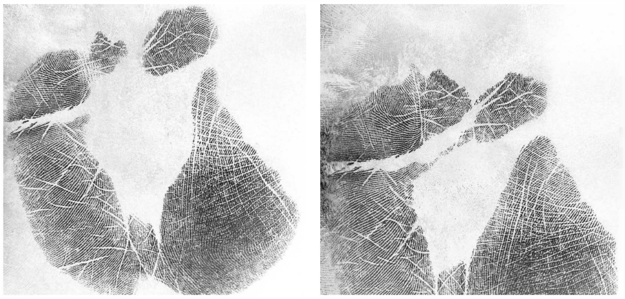


Figure 10. An example of failed retrieval

4.4. Indexing Efficiency

Given a query palmprint, the following steps are performed: feature extraction, multi-reference based pre-alignment, and matching score calculating. The time cost of the feature extraction algorithm [9] is about 22 seconds per palm. Then k (the pre-decided number of reference palmprints) indices of the query palmprint are created by aligning it to k reference orientation fields, each alignment taking about 1 second. As suggested in the experiment results in Table 1, we extract the feature vectors at the block size of 256×256 pixels and the orientation field and density map of a palmprint are both represented as 64-dimensional feature vectors. Calculating the orientation field similarity and the density map similarity between a pair of palmprints takes only 0.42×10^{-3} milliseconds and 0.44×10^{-3} milliseconds respectively. For $k = 20$, the average matching time between a pair of palmprints is 17.2×10^{-3} ($= 20 \times (0.42 + 0.44) \times 10^{-3}$) milliseconds. Performing a query on the whole database consisting of 12,716 palmprints (*i.e.* calculating the similarities between the query and all palmprints in the database) takes only 0.22 seconds for the proposed indexing algorithm. The software implementing the proposed algorithm has been made available for comparison purpose at `http://ivg.au.tsinghua.edu.cn/people/~JianjiangFeng/software.html`.

5. Conclusions and Future Work

Palmprint based person identification is receiving increasing attention from law enforcement community. A few palmprint matching systems have been proposed recently. However, the efficiency of these systems is far from satisfactory. In this paper, a ridge feature based palmprint indexing algorithm is proposed for large scale palmprint identification applications. The proposed indexing algorithm contains two key components: multi-reference based palmprint pre-alignment and orientation field and density map based coarse matching. The proposed indexing system achieved an error rate of 1% at the penetration rate of 2.25%. Searching a query palmprint takes only 0.22 seconds over a palmprint database consisting of 13,416 palmprints.

Current work can be extended in the following directions:

1. Fusion of multiple pre-alignment approaches. While the proposed reference orientation field based pre-alignment algorithm is robust in most cases, it cannot handle some partial palmprints. Combining the current approach with other pre-alignment approaches [4, 21] may improve the pre-alignment performance in difficult cases.

2. Learning based similarity measure. Similarity or distance measure between orientation fields and density maps can be learnt from training samples to better distinguish between genuine and impostor.

3. Minutiae based indexing. Minutiae set is the most distinctive and compact representation of palmprints. Combining an efficient minutiae-based indexing algorithm with the current algorithm may further improve the indexing performance.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants 61020106004, 60875017, 61005023, and 61021063.

References

[1] The FBI's next generation identification (NGI). *http://fingerprint.nist.gov/standard/presentations/archives/NGI_Overview_Feb_2005.pdf*, 2008.

[2] D. Ashbaugh. Quantitative-qualitative friction ridge analysis: An introduction to basic and advanced ridgeology. *CRC Press*, 1999.

[3] D. H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.

[4] R. Cappelli. Fast and accurate fingerprint indexing based on ridge orientation and frequency. *to appear on IEEE Transactions on Systems, Man and Cybernetics - Part B*.

[5] R. Cappelli, A. Lumini, D. Maio, and D. Maltoni. Fingerprint classification by directional image partitioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):402–421, 1999.

[6] J. Dai and J. Zhou. Multifeature-based high-resolution palmprint recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):945–957, 2011.

[7] S. Dewan. Elementary, watson: Scan a palm, find a clue. *The New York Times*, 2003.

[8] L. Fang, M. K. Leung, T. Shikhare, V. Chan, and K. F. Choon. Palmprint classification. *IEEE International Conference on Systems, Man and Cybernetics*, pages 2965–2969, 2006.

[9] A. K. Jain and J. Feng. Latent palmprint matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(7):1032–1047, 2009.

[10] A. K. Jain, L. Hong, and R. Bolle. On-line fingerprint verification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):302–314, 1997.

[11] A. K. Jain, A. Ross, and S. Prabhakar. An introduction to biometric recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1):4–20, 2004.

[12] X. Jiang, M. Liu, and A. C. Kot. Fingerprint retrieval for identification. *IEEE Transactions on Information Forensics and Security*, 1(4):532–542, 2006.

[13] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: an Introduction to Cluster Analysis.* John Wiley & Sons, Inc, 1990.

[14] A. Lumini, D. Maio, and D. Maltoni. Continuous vs exclusive classification for fingerprint retrieval. *Pattern Recognition Letters*, 18(10):1027–1034, 1997.

[15] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar. *Handbook of Fingerprint Recognition.* New York: Springer, 2006.

[16] NSTC Subcommittee on Biometrics. Palm print recognition. *http://www.biometrics.gov/Documents/PalmPrintRec.pdf*, 2009.

[17] Z. Sun, T. Tan, Y. Wang, and S. Li. Ordinal palmprint represention for personal identification. *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:279–284, 2005.

[18] X. Wu, D. Zhang, K. Wang, and B. Huang. Palmprint classification using principal lines. *Pattern Recognition*, 37(10):1987–1998, 2004.

[19] N. Yager and A. Amin. Coarse fingerprint registration using orientation fields. *EURASIP Journal on Applied Signal Processing*, 2005(13):2043–2053, 2005.

[20] D. Zhang, W. Kong, J. You, and M. Wong. Online palmprint identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1041–1050, 2003.

[21] D. Zhang and W. Shu. Two novel characteristics in palmprint verification: datum point invariance and line feature matching. *Pattern Recognition*, 32(4):691–702, 1999.