A New Fingerprint Indexing Algorithm for Latent and Non-latent Impressions Identification

José Hernández-Palancar and Alfredo Muñoz-Briseño(✉)

Advanced Technologies Application Center (CENATAV), Havana, Cuba
{jpalancar,amunoz}@cenatav.co.cu

Abstract. In this work, a new fingerprint identification algorithm for latent and non-latent impressions based on indexing techniques is presented. This proposal uses a minutia triplet state-of-the-art representation, which has proven to be very tolerant to distortions. Also, a novel strategy to partition the indexes is implemented, in the retrieving stage. This strategy allows to use the algorithm in both contexts, criminal and non-criminal. The experimental results show that in latent identification this approach has a 91.08% of hit rate at penetration rate of 20%, on NIST27 database using a large background of 267000 rolled impressions. Meanwhile in non-latent identification at the same penetration rate, the algorithm reaches a hit rate of 97.8% on NIST4 database and a 100% of hit rate on FVC2004 DB1A database. These accuracy values were reached with a high efficiency.

Keywords: Fingerprint indexing · Index partition · Polygonal hull

1 Introduction

Biometrics is the science of identifying people from particular physical features such as voice, fingerprints, iris texture or facial structure. One of the techniques used by biometric systems is the comparison of fingerprints due its uniqueness. Depending on the context of implementation of fingerprint recognition systems, two general classes of problems can be distinguished: verification (1 vs 1) and identification (1 vs many). A first approach to the identification of a person, could be to compare the given fingerprint with every one stored in the database. However, the size of current fingerprint databases is in the order of millions of impressions, so this approach is impracticable and some techniques to reduce the search space are needed, *e.g.*, indexing techniques. The methods based on indexing techniques return a list sorted by relevance of potential candidates to match with a given query. An additional complexity is presented when the query is performed using a latent fingerprint. These impressions are taken from crime scenes and generally the images have very bad quality.

In the literature, diverse approaches of indexing algorithms can be found. The differences between these methods are mainly in the selection of the features and in the indexing and retrieving stages. One of the most used features

A. Pardo and J. Kittler (Eds.): CIARP 2015, LNCS 9423, pp. 127–134, 2015.
DOI: 10.1007/978-3-319-25751-8_16

selection strategies is based on choosing triplets of minutiae [1–5] or other more complex geometric structures [6]. Other algorithms extract features directly from ridges [7], from a neighborhood around the each minutia [8,9] or from orientation maps extracted from every fingerprint [10]. On the other hand, there are few works about latent fingerprint indexing algorithms. Among them, the most relevant is an approach introduced by Paulino et al. [11]. In this work, a fusion of many features of level 1 and 2, like MCC [8], singular points and ridge periods, is performed in order to build a candidates list. Also, there is a proposal in the literature [12] that can be used for latent and non-latent impressions and uses polygons matching.

In the present work, a minutia based indexing algorithm is introduced. This proposal uses a previously defined representation of fingerprints [3] based on minutia triplets, which is very tolerant to distortions. There are many differences of our approach regarding the algorithm proposed in [3], among them is the use of other combination of characteristics and a novel strategy to partition the features and indexes. The partition strategy proposed by us allows to search latent and non-latent impressions on the same fingerprint database.

This work is organized as follows. In Section 2 the used representation and features are described. Also, the index construction and partition is defined. The Section 3 is dedicated to the indexing and retrieving stages using the primary and secondary indexes. In Section 4 experiments performed on public databases that validate our proposal are shown. Finally, Section 5 contains the conclusions.

2 Indexes Generation

In order to perform a proper identification, from each fingerprint some indexes representing characteristic information must be extracted. These indexes are conformed by features extracted from the used representation and in posteriors stages they can be used for finding correspondences.

2.1 Fingerprint Representation

The minutia is the most common feature used in fingerprint recognition algorithms. Minutiae are singularities in the ridges pattern, which are commonly used by experts for performing manual comparisons. In latent fingerprint case an expert manually marks the minutae. In this way, some indexing approaches use minutia triplets in order to represent impressions [1–5]. In this work, is used one of these representations, which has proven to be very effective for situations in which some minutiae are not located correctly [3]. This approach defines an expanded triplets set $R = \{t_1, t_2, \cdots, t_n\}$ in which the vertexes of each triplet are conformed by minutiae.

The number of triplets of R is linear with respect to the number of minutiae [3]. This is very desirable since the sets R will be used as a representation for fingerprints in indexing tasks. This property also has the advantage that the identification errors by false acceptance are reduced in comparison with other approaches that use all possible triplets [2] or only a Delaunay triangulation [1].

2.2 Features Used

In the present work, a study of the different features that can be extracted from a triplet of minutiae present on the state-of-the-art, was conducted. As result, the better features for fingerprint recognition in both cases, civil and criminal, were identified. In order to generate indexes, from each $t_i \in R$, where $p_i(x_i, y_i)$, $p_j(x_j, y_j)$ and $p_k(x_k, y_k)$ are the points of t_i , the following features are extracted attending to their robustness:

- s_t: Triangle sign, where $s_t = 0$ if $x_i(y_j - y_k) + x_j(y_k - y_i) + x_k(y_i - y_j) < 0$; otherwise $s_t = 1$.
- θ_i, θ_j and θ_k: Normalized difference of directions of minutiae represented by p_i, p_j and p_k, with respect to their opposite side on t_i.
- L_1 and L_2: Normalized heights of the triplets. Heights of the smallest and the largest side of each t_i with respect to the opposite point in the triplet
- r_i, r_j and r_k: Ridge counters.
- ρ_i, ρ_j and ρ_k: Relative position of p_i, p_j and p_k regarding a reference point.

The order of the minutiae in a triplet is given by the sides length of the triangle that they describe. The features were obtained from the minutiae extracted with VeriFinger 4.2. In the performed studies, was observed that the min-max heights regarding each triangle side although are not easy to calculate, provide a very stable feature. Also, was concluded that the relative position regarding a reference point avoids false correspondences between impressions.

2.3 Index Generation and Partition

Since in this work is introduced an algorithm that can be used with latent and non-latent impressions, and the features may have different identification value in correspondence with this fact, a partition of the index function is proposed, in Primary (PI_i) and Secondary (SI_i) indexes, as can see in Table 1. In this way, PI_i is used in the presence of both, latent and non-latent impressions.

Table 1. Binary representation of triplet features used.

Index part	Feature	Size (bits)
Primary	s_t	1
	$\theta_i, \theta_j and \theta_k$	$3+3+3=9$
	$L_1 and L_2$	$3+3=6$
Secondary	r_i, r_j and r_k	$6+6+6=18$
	ρ_i, ρ_j and ρ_k	$3+3+3=9$

The indexes are constructed by concatenating the binary representation of each involved feature. In Table 1 the amount of bits necessary for representing the used features based on the higher value that each one can reach is shown. As can be deduced from this, each index can be stored in a very compact way.

3 Indexing and Retrieving Processes

In this section, a detailed description of the process of identification is made. For this, a very efficient technique that uses the previously defined primary and secondary indexes is used.

3.1 Indexing

This process is made in preprocessing time, when a fingerprint with an assigned ID is inserted in the database. The first step is the construction of its expanded triplets set R. Then, from each one of these triplet $t_i \in R$, all the mentioned features are extracted and from these, PI_i and SI_i are made up. With this information, a tuple $(PI_i, SI_i, ID, t_i(p_1p_2p_3))$ is built from each t_i. Finally, every one of these tuples is inserted in an index table H, using PI_i as primary index. In this way, H will contain information about all the triplets generated from every fingerprint inserted in the database. It is important to note that H allows to store more than one element under the same key (primary index). In our case, this is very desirable since some triplets may generate the same primary index. However, these triplets may have different secondary index and ID.

3.2 Retrieving

In this stage, a similarity value is computed between the query fingerprint and every impression stored. Then, the candidates list is made up by the elements from the database that archive a similarity value higher than 0, ordered in decreasing way. In order to perform this operation, the expanded triplets set R_q is computed from the query. In this way the primary and secondary indexes PI_{iq} and SI_{iq} from each $t_{iq} \in R_q$ are computed. Then, a query is performed on the index table H using all the PI_{iq} as keys. If we are not in the presence of a latent impression, those elements stored in H that have a secondary index different than SI_{iq} are discarded. On the contrary, if the query is a latent impression, then this second level filtering is not used. As we can see, this strategy allows us to solve both cases in a very efficient manner.

As result of the previously described operation, n tuples with an associated identifier ID_i and with the form $R_{ti} = \{t_1, \ldots, t_k\}$ can be obtained. These tuples are made up from the elements retrieved from H, grouping those triplets that contain the same ID. After this process, a triplets matching strategy very similar to one already defined in the literature can be used for matching the sets R_q and R_{ti} [13]. The correlation tuples between two triplets $t_q(p_{1q}, p_{2q}, p_{3q})$ and $t_t(p_{1t}, p_{2t}, p_{3t})$ are defined as $ct_i = (\alpha_i, \overline{p_{iq}p_{jq}}, \overline{p_{it}p_{jt}})$ with $i, j \in \{1, 2, 3\}$, were α_i represents the normalized difference between the i-th interior angles of t_q and t_t, and $\overline{p_{iq}p_{jq}}, \overline{p_{it}p_{jt}}$ are segments of the triangles represented by the triplets. The process followed to obtain the value of α_i, is very similar to that presented by Chikkerur *et al.* [9], to obtain the similarity between an edge that connects two minutiae of an impression and one edge joining two minutiae of other fingerprint. In this way, the set $T(R_q, R_{ti}) = \{ct_1, ct_2, \ldots, ct_n\}$ represents the union of all the correlation tuples of every corresponding triangle between R_i and R_j. In the following step, a reduced set $Tr(R_q, R_{ti}) = \{ct_1, ct_2, \ldots, ct_r\}$ is used. This set contains only the correlation tuples whose value of α_i are equal to the statistic mode in $T(R_q, R_{ti})$, for the values of α_i of every ct_i. The main goal of this process is to check the rotation histogram for finding the most probable value of relative rotation between the involved feature models and use only the correlation tuples that are consequent with this.

With the reduced set $Tr(R_q, R_{ti})$ a similarity graph $G_s = \langle V, E, L, s, l \rangle$ is made up where $s : E \rightarrow \Re$ is a similarity function that assign a value to every edge, $l : {m_i}^2 \rightarrow L$ is a labeling function given two vertexes and L is a set of vertex labels. s is a similarity function that represents in fuzzy terms the grade of closeness between the two segments $\overline{p_{iq}p_{jq}}$ and $\overline{p_{it}p_{jt}}$ that originated a edge in G_s. In this way, a graph that represents matches between triplet sets R_q and R_{ti} is constructed and weighted with a similarity function. The graph G_s may be not connected. More details of the construction of G_s can be found in a previous work found in the literature [13].

In order to find the spanning tree of every connected components of G_s with the higher value of similarity in their edges, the Kruskal algorithm is applied to G_s. This is a well know method to find a minimum (or maximum in our case) spanning forest of disconnected graphs. Let $\{F_1, F_2, \ldots, F_n\}$ be the set of spanning trees returned by the Kruskal algorithm, sorted in descending order by the amount of edges. A strategy to merge F_1 and F_2 is implemented by trying to add a virtual edge e_v between then. This virtual edge must complain with some geometric restrictions. If this process is successful then $F_1 = F_1 \cup F_2 \cup \{e_v\}$, F_2 is eliminated and $F_i = F_{i-1}, \forall\ i, 3 < i < n$. This process is repeated while F_1 and F_2 can be merged.

The similarity value between the triplets sets R_q and R_{ti} is given by the following expression:

$$similarity(R_q, R_{ti}) = \frac{sim \times |V|}{min(|R_q|, |R_{ti}|)} \tag{1}$$

where $|V|$ is the number of vertexes in the similarity graph G_s, $|R_q|$ and $|R_{ti}|$ are the cardinalities of R_q and R_{ti} respectively, and sim is the sum of the weights of every edge of F_1. Finally, with the process described, a similarity score can be generated between the representation of the query R_q and every impression previously stored in the database that shares correspondences with the query. The final candidates list can be obtained by sorting these impressions by the computed score. In Fig. 1 the flow of the algorithm is shown. As can be seen, the use of primary and secondary indexes provides a great flexibility to this proposal. It is important to note that this work uses an efficient consolidation strategy that does not compromise the response times and that is similar to the used in [13]. This does not mean that we use a matching algorithm, because comparisons 1:1 are not performed, so the results shown are based on an indexing scheme.

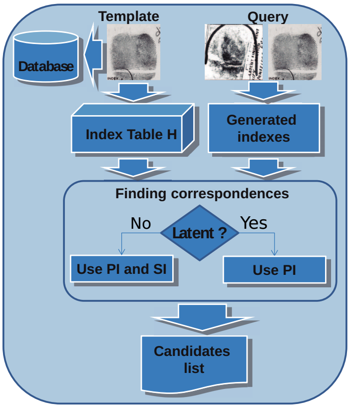


Fig. 1. Flow of the fingerprint identification process.

4 Experimental Evaluation

In this section, an evaluation of the accuracy of our proposal is performed, compared with other state-of-the-art algorithms. For this, the most common measure to evaluate the accuracy of indexing algorithms was used, i.e., the trade off between penetration rate (PR) and Correct Index Power (CIP). The experiments were executed on a PC with a microprocessor i7, 1.7 Ghz and 8 Gb of RAM.

4.1 Latent Case

The NIST Special Database 27 was used to test the accuracy of our proposal. This database contains 258 latent fingerprints from crime scenes and their matching rolled impressions mates. For each latent fingerprint there are four sets of minutiae that have been validated by a professional team of latent examiners. From these, only the set of manually marked minutiae called ideal was used. Also, a large background database of 267000 rolled impressions obtained from NIST databases and a private database of our country was used, in order to fairly compare our results with others works [11,12]. As can be seen in Fig. 2, our proposal outperforms other approaches for almost all values of penetration rate. In particular, we can see that the better results of our algorithm compared to other methods, are reached for a penetration value of 1%. In criminal cases, the searches are performed on the whole database and the decisions are made by human experts. That is why it is important that the searched candidate is included on the firsts positions of the returned candidates list. In latent case this approach was able to perform approximately 50000 comparisons per second.

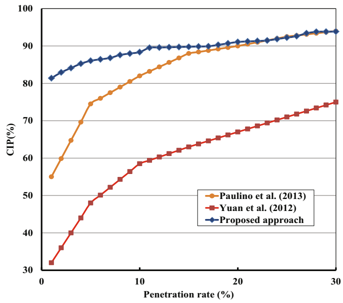
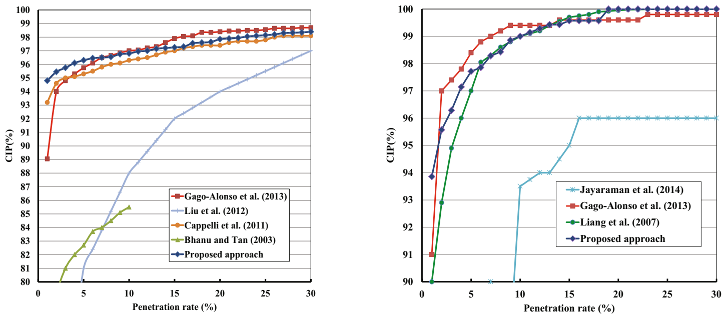


Fig. 2. Experiments performed in the NIST 27 database.

4.2 Non-latent Case

Two well-known databases were used for testing the accuracy of our approach. The first one is FVC2004 DB1 that contains 800 fingerprints (100 fingers and 8 impressions per finger). The second one was the NIST Special Database 4, composed by 4000 rolled impressions (2000 fingers and 2 impressions per finger). The Figures 3(a) and 3(b) show that in general the approach results are very good, and in particular are better than the others proposals for the first values of penetration rate. The algorithm was able to perform 1.5 millions comparisons per second, for this test we used a variation of the same background database employed in latent case.



(a) NIST 4 (b) FVC2004

Fig. 3. Experiments performed in different databases.

5 Conclusions

The proposed indexing algorithm introduces a very efficient strategy to identify latent and non-latent fingerprints, implementing two search levels based on the use of a primary index and a secondary index, allowing the searching of latent and non-latent impressions on the same fingerprint database, due to this, it is possible to use the same algorithm for civilian and forensic applications. The features used for index generation and fingerprint representation have a great robustness in the presence of noise. Our method outperforms some of the best approaches in state-of-the-art, reaching a good hit rate at low values of the penetration rate. This is a very desirable characteristic for identification systems since minimizes the response time and was one of our goals in the development of this work. The proposed approach performs 50000 and 1.5 millions comparisons per second for latent and non-latent cases respectively. The cited approaches do not report execution times, so comparisons were not performed.

References

1. Bebis, G., Deaconu, T., Georgiopoulos, M.: Fingerprint identification using delaunay triangulation. In: IEEE International Conference on Intelligence, Information, and Systems (ICIIS 1999), pp. 452–459 (1999)
2. Bhanu, B., Tan, X.: Fingerprint indexing based on novel features of minutiae triplets. IEEE Transactions on Pattern Analysis and Machine Intelligence **25**(5), 616–622 (2003)
3. Gago-Alonso, A., Hernández-Palancar, J., Rodríguez-Reina, E., Muñoz-Briseño, A.: Indexing and retrieving in fingerprint databases under structural distortions. Expert Syst. Appl. **40**(8), 2858–2871 (2013)
4. Khachai, M.Y., Leshko, A.S., Dremin, A.V.: The problem of fingerprint identification: A reference database indexing method based on delaunay triangulation. Pattern Recognit. Image Anal. **24**(2), 297–303 (2014)
5. Liang, X., Bishnu, A., Asano, T.: A robust fingerprint indexing scheme using minutia neighborhood structure and low-order delaunay triangles. IEEE Transactions on Information Forensics and Security **2**(4), 721–733 (2007)
6. Iloanusi, O.N.: Fusion of finger types for fingerprint indexing using minutiae quadruplets. Pattern Recogn. Lett. **38**, 8–14 (2014)
7. Feng, J., Cai, A.: Fingerprint indexing using ridge invariants. In: Proceedings of the 18th International Conference on Pattern Recognition, ICPR 2006, Washington, DC, vol. 04, pp. 433–436. IEEE Computer Society (2006)
8. Cappelli, R., Ferrara, M., Maltoni, D.: Fingerprint indexing based on minutia cylinder-code. IEEE Trans. Pattern Anal. Mach. Intell. **33**(5), 1051–1057 (2011)
9. Chikkerur, S., Cartwright, A.N., Govindaraju, V.: K-plet and coupled BFS: a graph based fingerprint representation and matching algorithm. In: Zhang, D., Jain, A.K. (eds.) ICB 2005. LNCS, vol. 3832, pp. 309–315. Springer, Heidelberg (2005)
10. Liu, M., Yap, P.T.: Invariant representation of orientation fields for fingerprint indexing. Pattern Recogn. **45**(7), 2532–2542 (2012)
11. Paulino, A.A., Liu, E., Cao, K., Jain, A.K.: Latent fingerprint indexing: fusion of level 1 and level 2 features. In: Biometrics: Theory, Applications and Systems (BTAS), Washington, D.C., pp. 1–8 (2013)
12. Yuan, B., Su, F., Cai, A.: Fingerprint retrieval approach based on novel minutiae triplet features. In: Biometrics: Theory, Applications and Systems (BTAS), Arlington, VA, pp. 170–175 (2012)
13. Hernández-Palancar, J., Muñoz-Briseño, A., Alonso, A.G.: Using a triangular matching approach for latent fingerprint and palmprint identification. IJPRAI **28**(7) (2014)