State of the Art of Fingerprint Indexing Algorithms

Estado del arte de algoritmos de indexación de impresiones dactilares

Alfredo Muñoz Briseño, Andrés Gago Alonso, and José Hernández Palancar
Advanced Technologies Application Center, 7a #21812 e/ 218 y 222, Siboney, Playa,
Havana, Cuba
{amunoz, agago, jpalancar}@cenatav.co.cu

Article received on March 23, 2011; accepted on June 30, 2011

Abstract. Due to the large size that fingerprint databases generally have, the reduction of the search space is indispensable. In the resolution of this problem, indexing algorithms have a fundamental role. In the literature, there are several proposals that make use of different features to characterize fingerprints. In addition, a wide variety of recovery methods are reported. This paper concisely describes the indexing algorithms that have reported better results so far and makes a comparison between these, based on experiments in well known databases. Finally, a classification of the indexing algorithms is proposed, based on some general characteristics.

Keywords. Indexing algorithms, fingerprints verification, fingerprints features, triplets features and ridges features.

Resumen. Debido al gran tamaño que pueden alcanzar las bases de datos de impresiones dactilares, se hace indispensable la reducción de espacio de búsqueda. En la resolución de este problema, los algoritmos de indexación juegan un papel fundamental. En la literatura sobre el tema, existen algunas propuestas que hacen uso de diferentes rasgos para caracterizar las impresiones. Además, existen reportados una gran variedad de métodos de recuperación. El presente artículo describe de manera concisa, los algoritmos de indexación que han reportado los mejores resultados hasta ahora y se hace comparaciones entre estos, basados en experimentos en bases de datos conocidas. Finalmente, se propone una clasificación, basada en algunas características generales.

Palabras clave. Algoritmos de indexación, verificación de impresiones dactilares, rasgos de impresiones dactilares, rasgos de tripletas y rasgos de crestas.

1 Introduction

Biometrics is the science of identifying people from particular physical features such as voice, fingerprints, iris texture or facial structure [5]. One of the techniques used by biometric systems is the comparison of fingerprints. The patterns of ridges found on fingers and other body parts are unique and provide enough information to distinguish a specific person from the rest.

Depending on the context of implementation of fingerprint recognition systems, we can distinguish between two general classes of problems: verification and identification. Since the purpose of verification systems is to confirm the identity of a particular individual, comparisons are only made with fingerprints stored that belong to that person. On the other hand, the identification is more complex because it requires a search on all fingerprints stored in the database. A first approach to the identification of a person, could be the comparison to the given fingerprint with all the stored in the database. However, the size of most modern databases is in the order of millions of impressions, so this method is impracticable.

A referred solution in the literature is the classification of fingerprints according to the five classes of Henry [3] (Arch, Left loop, Right Loop, Whorl and Tented Arch). These classes divide the impressions into groups based on patterns formed on the ridges. In this way the search is reduced to the subset of stored impressions which share the same class as the query. However, this method has serious disadvantages mainly because the number of classes to divide the search space is small. In addition 90% of impressions belong to three classes, so, in most cases, the reduction of potential candidates is insignificant.

Another solution, called indexing, is used to solve this problem. The methods based on this technique are capable of, given the value of a key, returning a list sorted by relevance of potential candidates to match.

This work concisely describes the indexing algorithms that have achieved better results until today, taking into account accuracy and efficiency. In this study, some of the most referenced algorithms are classified, according to the fingerprint representation strategy used for indexing.

The rest of this paper is organized as follows. Section 2 presents a state of art of the major indexing algorithms currently available. Section 3 describes the experiments and comparisons and finally our conclusions are given in Section 4.

2 Indexing Algorithms

The basic structure of all indexing algorithms is the same. Given an impressions database defined as $D = \{f_1, \ldots, f_n\}$, where f_i represents the i-th stored impression and n is the number of impressions contained in D, each f_i is preprocessed before being introduced in D. The purpose of this is to extract certain features for calculating a set of indexes V_i for each f_i, which can help to reduce the search space in a fraction of D. As a result, all of these indexes are stored.

In the recovery process, when a query, denoted by f_q, is performed, the set of indexes V_q of f_q is calculated with the same method used to extract the sets V_i. Thus, a list of candidates L_c is obtained, formed by the impressions $f_i \in D$ whose indexes have more correlation with f_q. The correlation between f_q and a given f_i is expressed by a numeric value called index score, denoted by S_i. Figure 1 illustrates graphically the described process. Based on the representation method and the features selected of the fingerprint, we can classify the indexing algorithms in several categories (based on minutiae, based on ridges, based on transform and other approaches).

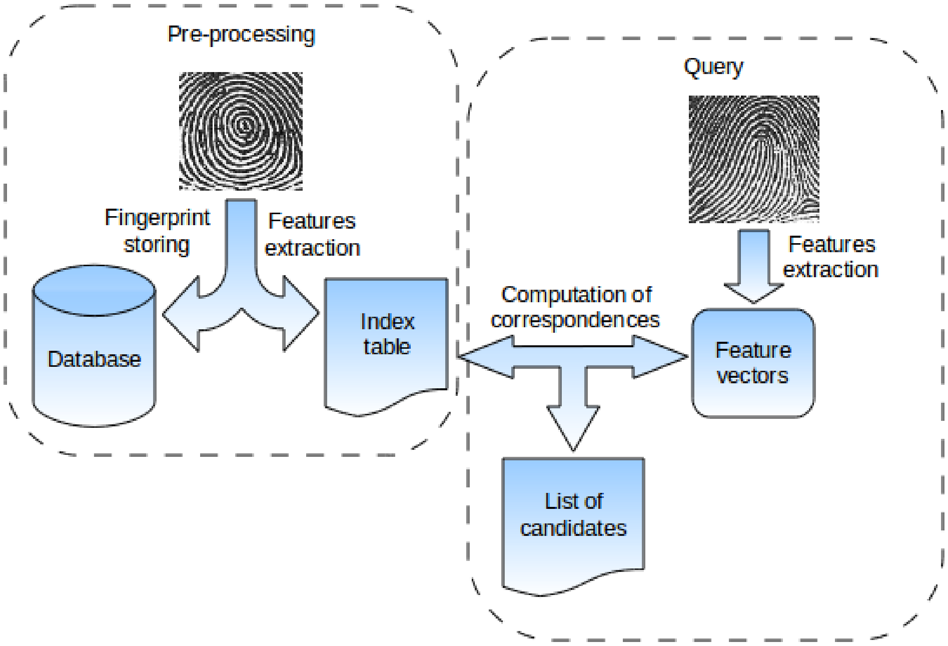


Fig. 1. Processes of indexing and recovery

2.1 Algorithms based on minutiae

Many reported indexing methods are based on the extraction of features in all triplets that can be formed between the minutae of the fingerprints.

Germain *et al.* [5] proposed an algorithm that introduces a new recovery method to generate L_c. This method uses two structures, called map and multimap, to compute all the S_i values. In preprocessing time, the extracted features from each f_i, are used to generate the set V_i. For each element of V_i, an entry is added to the multimap. In the recovery process, each element of the set V_q generated by a given f_q is used to retrieve any items in the multimap that are stored under the same index. These items are labeled with some geometric transformation that brings the subset of minutiae that generated the indices into closest correspondences. The map is used to calculate the number of obtained items, which have the same geometric transformations and were generated by the same fingerprint stored in the database. With these values, L_c is constructed. Figure 2 illustrates the process where I represents the generated indexes and Tr is a geometric transformation.

Bhanu and Tan [2] describe a variant that proposes some alternative features like direction and orientation of the triangles formed by the triplets or type of minutiae (bifurcation or termination). In the recovery process, the index score of a f_i with respect to a given f_q, is expressed by:

$$S_i = \sum_{j=1}^{N} r_j \quad (1)$$

where r_j is given by the number of matched triangles between f_q and f_i, and N represents the number of potential corresponding minutae. In the computation of S_i are considered only the fingerprints with value of r_j greater than a defined threshold.

If all possible triplets that can be formed in a given fingerprint are considered, a cubic $O(n^3)$ number of indexes is generated. This affects the memory requirements. In the literature, some strategies have been proposed for the selection of triplets to deal with this problem. One of them is to form triplets with minutiae that are closer than a defined threshold [5]. However, this solution could originate major losses of characteristic information. Another approach is to use triangulations, in order to associate unique topological structures with the minutiae sets that represent the fingerprint [1, 8, 10, 12].

One of the first algorithms in use Delaunay triangulations to form the triplets, was defined by Bebis *et al.* [1]. In this algorithm, once the triplets of a given f_i are obtained, for each of these, some features are extracted to form V_i. A recovery method similar to that described by Germain *et al*. [5] is used by this algorithm.

Mukherjee [12] suggests other features for calculating V_i vectors using the ridges associated with each minutia and the coefficients of his respective second-degree curves. Once the vectors of all triplets have been extracted, they are partitioned into k clusters that are going to be used as indexes, applying for this, the k-means algorithm. In the recovery phase, given a f_q, for each f_i the same process is performed. The S_i value is computed from the correspondence between the clusters generated by f_q and all f_i.

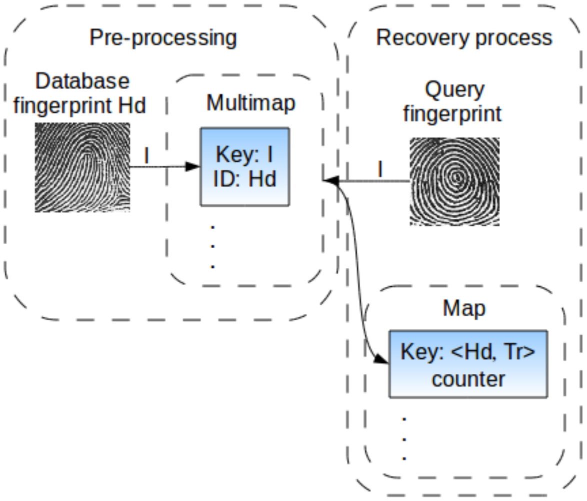


Fig. 2. Recovery process defined by Germain *et al.* where Hd is the fingerprint introduced in D

On the other hand, Liang *et al.* [11] propose several features already used by Bhanu and Tan [2] to form the V_i vectors. The authors also introduce a feature T_j^i that is given by the relative positions of segments b_1, b_2 and a of length λ, which are tangents to the ridges associated with each minutia that represents a bifurcation. Figure 3 illustrates the extraction of this feature. Another feature introduced is the difference between the angles of two edges associated with the analyzed minutia, and their respective directional field.

The used recovery method is similar to the defined by Bhanu *et al.* The same authors proposed an improved variant [10] using a Delaunay triangulation of higher order, to increase the number of triplets and consequently the total of indexes computed. Table 1 shows briefly some of the characteristics of the better known indexing algorithms based on minutiae.

2.2 Algorithms Based on Ridges

In the currently available literature, we can find some algorithms that use the ridges of fingerprints as a means to represent them. Some of them used second-degree polynomials to simulate the curves of the ridges, which can be used to extract features [2, 12].

Biswas *et al.* [3] introduced a variant of this idea from collecting information on the ridges in a given neighborhood of each minutia. Specifically, the representation of the fingerprint is divided in oriented blocks of $50x50$ pixels around each minutia. Thus, a feature formed by the minutia analyzed and the average of the two highest values of the curvatures of the ridges in that block is extracted. The authors also described a feature vector to classify the fingerprints in Henry classes, which reduces the search space.

Feng and Cai [6] introduced another way to use the information that provide the ridges to the indexing process. The authors proposed the creation of substructures formed by the ridges that converge in each minutia. For those who represent bifurcations, the associated substructure consists in the three ridges that converge in the minutiae. For those who represent terminations, besides the ridge that ends, the two adjacent are saved and each of these is divided into two sub ridges.

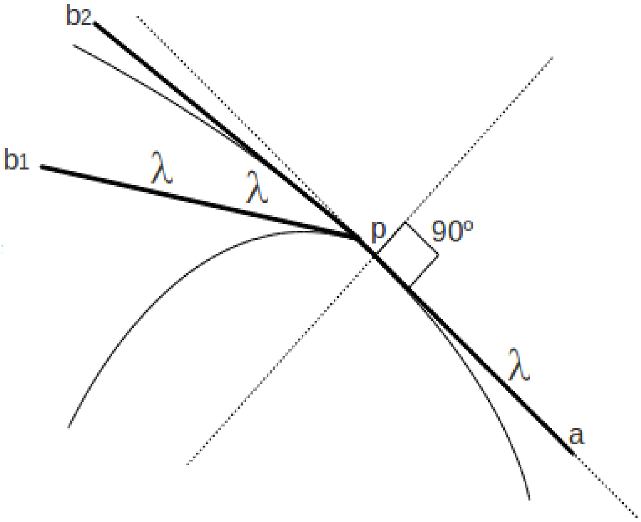


Fig. 3. Extraction of the feature defined by Liang *et al.*

Table 1. Characteristics of triplets based algorithms

Algorithm	Used features	Triplets formation	Correspondences computation
Germain *et al.* [5]	- lengths of the sides - amplitudes of the angles - ridge counters	All triplets	Use map and multimap structures
Bhanu and Tan [2]	- amplitudes of angles - triangle orientation - type of minutiae (2 cases) - maximum side length - triangle direction	All triplets	Number of correspondences between triplets
Bebis *et al.* [1]	- lengths of the sides - amplitude of maximum angle	Delaunay triangulation	Similar to Germain *et al.* [5]
Liang *et al.* [9]	- triangle orientation - amplitudes of angles - maximum side length - type of minutiae (10 cases)	Delaunay triangulation	Similar to Bhanu and Tan [2]
Liang *et al.* [10]	- triangle orientation - amplitudes of angles - maximum side length - type of minutiae (10 cases)	Higher order Delaunay triangulation	Similar to Bhanu and Tan [2]
Mukherjee [12]	- amplitudes of angles - lengths of the sides - amplitudes of angles - second-degree curve coefficients	Delaunay triangulation	Correspondence between the query clusters and the clusters stored in the database

Once the substructures have been obtained, the ridges and sub ridges are labeled according to their relative positions with respect to the analyzed minutia. The ridges adjacent to the minutiae are divided starting from a line perpendicular to the direction of the minutia. The indexes are derived from binary relations between substructures and the labels generated by its associated ridges. Given a query f_q,it evaluates the number of correspondences with the substructures for all f_i stored, in order to generate the S_i values. Figure 4 shows the defined substructures.

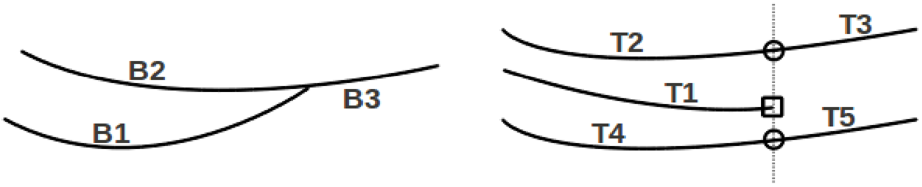


Fig. 4. Substructures of ridges defines by Feng and Cai [6]

2.3 Transform Based Algorithms

Image processing is any form of processing in which the output may be another image, a set of characteristics or parameters related to the image. In the specific case of fingerprint indexing, there are some methods defined that make use of filters to extract features [7, 8].

Kumar [7] uses Gabor filters to obtain the index vectors. The author proposal is to consider only the minutiae with reliability greater than a defined threshold for features extraction. Gabor filters are applied in neighborhoods of each minutiae from which V_i vectors are obtained. These are partitioned into K clusters, using the K-means algorithm. In turn, each K is partitioned into k clusters to further reduce the search space. Finally, the indexes are generated for each cluster k using R-tree structures.

In the recovery process of a query f_q, their respective V_i vectors are obtained and for each of these its corresponding cluster is found, by doing a search that minimizes the Euclidean distance between all the clusters previously generated. For each V_i, the R-trees generate tuples from which the S_i values are computed.

On the other hand, Li *et al.* [8] use symmetric filters from cores, deltas and parallel patterns. The authors propose applying the three filters to the orientation image obtained from the f_i stored. As result of this, three vectors V_i are obtained from the response matrices. These are reduced to a fixed length and rotated to set the direction of the cores as a vertical line.

When a query f_q is performed, three lists of candidates are generated starting from the similarity of the Euclidean distance values between the extracted index vectors of f_q and all f_i. The resulting lists are combined to form the final candidate list L_c.

2.4 Other Approaches

There are some algorithms that exploit the features offered by characteristic points, obtained in various ways. Liu proposed an algorithm that obtains the singular points of the query, based on directional fields [11]. For this, a “T-shape model” is defined which also allows the extraction of the directions of the singular points. The V_i vectors and S_i values are computed by applying MACE filters to defined neighborhoods around the singular points of various impressions of a same finger.

Shuai *et al.* also introduces an algorithm that extracts characteristic points of impressions, based in SIFT descriptor [13]. These are much more numerous than those resulting from the extraction of minutiae. In addition, the number of SIFT features can be reduced by decreasing the scale factor as much as desired. The author proposed the selection of the most significant characteristic points, for the extraction of the features, taking into account the value of contrast. This is done to obtain a stable number of features.

Finally, the authors proposed a locality-sensitive hashing (LSH) algorithm to obtain only one set of features, given three impressions from the same finger.

One idea to deal with the disadvantages of some indexing algorithms is to combine several of these at the time of generating the candidate lists. Based on this, Boer *et al.* [4] proposed to combine three indexing methods which use different representations of fingerprints.

The first method splits the impressions in blocks of $16x16$ pixels and extracts values that represent the local direction of the ridges, in each block. All these values are concatenated in a vector, whose dimension is reduced by applying a Karhunen transform. The results of these operations are used to build L_c.

The second method generates fixed length V_i vectors, starting from the FingerCodes of the impressions. This is calculated by applying Gabor filters, to capture the global and local features of the fingerprint.

The last method is based on features and invariants extracted from triplets. With these data, V_i vectors are computed and L_c is generated following a very similar idea to the described by Bhanu and Tan [2]. Finally, two ways of merging the three lists of candidates obtained are proposed.

3 Comparison among Described Algorithms

Several aspects must be taken into account in an indexing algorithm. One of the most important ones is how to extract the indexes. For acceptable performance, these should be calculated from features of the impressions that have minimal variation in situations of noise or distortion. In this way, the correct identification is ensured, even if the quality of the images is low. Due to this reason, the results of some algorithms are good [5, 6]. In the algorithm defined by Feng *et al.* [6], the extracted features have little variation in situations of noise, since they are based on structural relations between the ridges. In addition, the number of indexes that can be found given three minutiae is much greater than any triplets based algorithm. Germain *et al.* [5] also introduces a feature that improves the performance of the proposed algorithm, this is the case of ridges counters, with error thresholds. On the other hand, Liang *et al.* [9, 10] defines a very robust feature that classifies bifurcations of the ridges in ten cases. This classification has the disadvantage that, in practice the amount of bifurcations types is not uniformly distributed.

Other algorithms make use of features such as side lengths and the amplitudes of the angles of the triplets extracted [1, 12]. These features have proven to be extremely susceptible to possible distortion of the impressions and therefore adversely affect the algorithms that use them.

Another important aspect is the number of indexes that are generated and the ability they have to capture the characteristics that best defines the fingerprints. This is a factor that negatively influences the performance of the algorithm proposed by Bhanu and Tan [2]. In this one, the indexes are resistant to noise, but all possible triplets that can be formed at the time of generating them are taken into account. Some algorithms make use of Delaunay triangulation to reduce the number of triplets [1, 9, 10]. The biggest problem with this method is that, by reducing the number of triplets generated, some characteristic information can be lost. The proposal of Liang to deal with this problem improves performance relative to other triangulations based algorithms. In Figure 5 we can see the different amount of triplets that are generated by considering all possible triplets 5 (a) or Delaunay triangulations 5 (b). The image used as example in Figure 5 belongs to the FVC 2004 DB1_A database.

On the other hand, some of the filter-based methods have the advantage that they can be applied to any type of image to calculate grades of similarity; however, the implementation of these is expensive due to the amount of information generated and the number of calculations needed. Also, some of these algorithms depend on the location of singular points [7, 6, 13].

This has a major problem in low quality impressions. In the algorithm defined by Shuai *et al.* [13], the authors make use of an image recognition method, which gives poor characteristic information in the specific case of fingerprints. Also, the value of contrast is not a reliable measure to ensure the selection of the most significant points.

In the case of the algorithm defined by Boer *et al.* [4], in one of the used algorithms, a lower performance than the reported by the original author is obtained . Although the combination of the lists of candidates is an interesting idea, one must be careful when selecting methods, because of the costs and possible disadvantages.

In many of the described algorithms [5, 2, 1, 13, 12, 9], experiments were performed by the respective authors on the FVC 2002 DB1_A database, from the established correlation between percent penetration rate and correct index power.

The correct index power of an algorithm is defined as the percentage of correct fingerprints found in the top positions of the list of candidates returned. On the other hand, the penetration rate (P) is the average percentage of fingerprints in the data base retrieved over all input fingerprints. More formally we can define P as:

$$P = \frac{n * 100}{N} \% \tag{2}$$

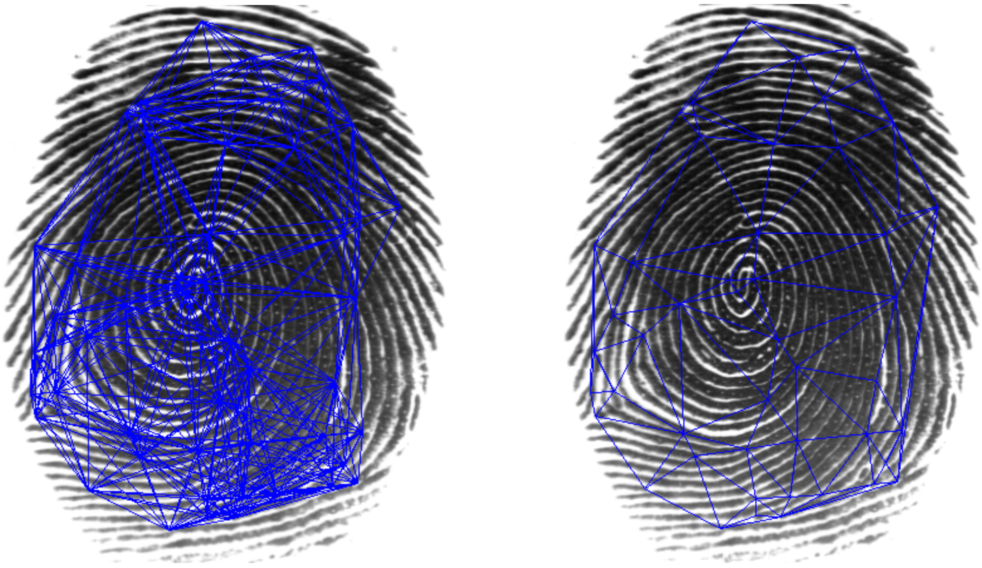
where n represents the number of accessed fingerprints of those returned by the indexing algorithm and N represents the total number of fingerprints contained in the database.

Figure 6 shows a comparison of the results reported or provided by the authors of the methods. Among the algorithms that use minutia, the best results are those reported by Germain *et al.* [5] when the penetration rates are up to 20%. But for higher values, the Bebis *et al.* proposal [1] is the most reliable. However, we can conclude that Feng and Cai algorithm [6] is much more robust than those who pose estimates based on triplets.

Also, Liang *et al.* algorithm [10] achieves very good results compared with other triplet based methods. This can be seen in Figure 6 (b), where also a comparison is made with the Shuai *et al.* [13] and Mukherjee [12] proposals.

The FVC 2002 DB1_A database is formed by 800 fingerprints, eight prints each of 100 distinct fingers. It is important to note that the methodologies used in the experiments described in Figure 6 (a) and Figure 6 (b) has some important variations, even when in both cases the same database is employed. All the described experiments were made by constructing the template database with n impressions of each finger, randomly chosen. The remaining 8-n impressions were taken as queries. The difference between the results shown, is that in the Figure 6 (a), $n = 1$ while in the Figure 6 (b), $n = 3$

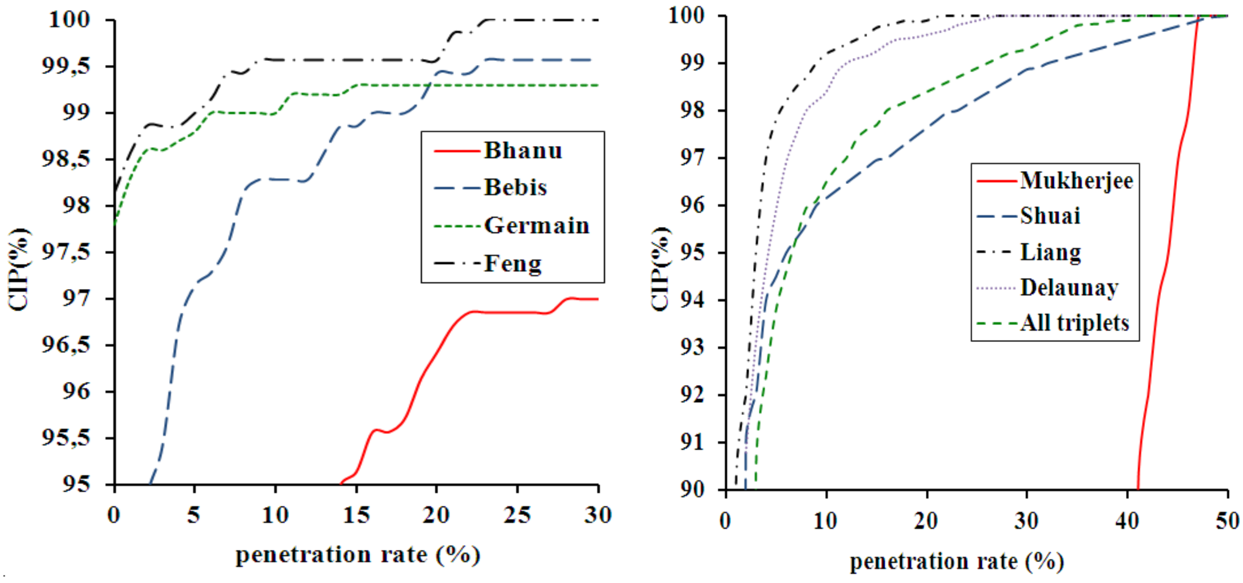
.



(a) (b)

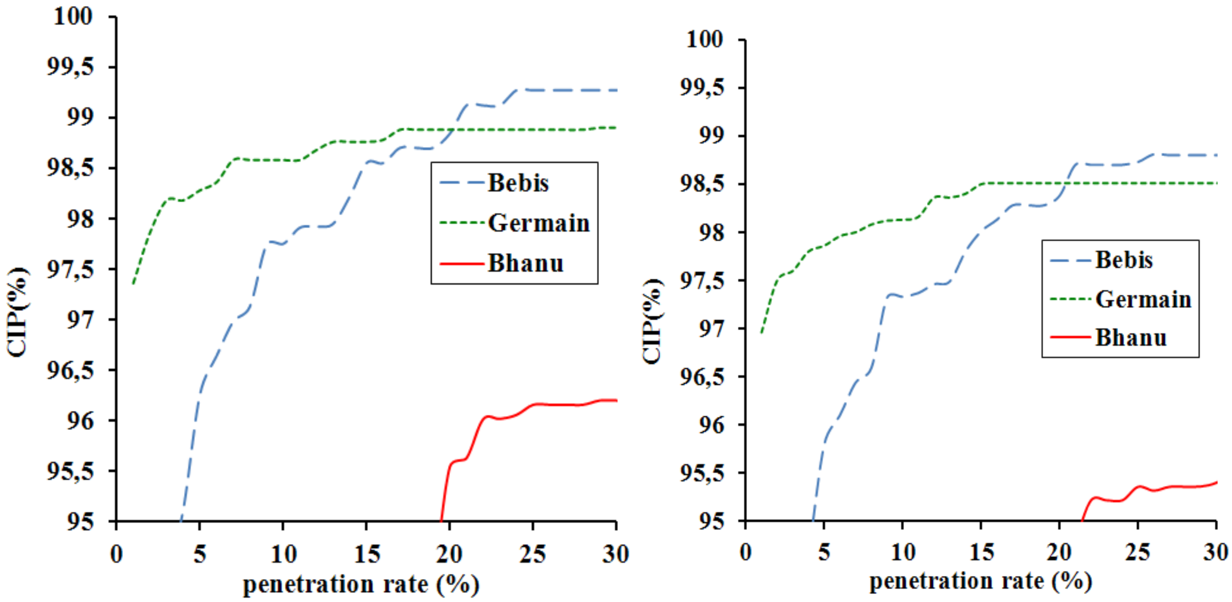
Fig. 5. Triplets generated with different algorithms

Since the methods that use triplets are the most mentioned in the literature, we have conducted experiments with some of them [1, 2, 5]. The methodology and parameters used were the same that the original authors of the algorithms employed, with $n = 1$. The experiments were performed on the FVC 2002 DB2_A and DB3_A databases. The results are shown in Figure 7.



(a) (b)

Fig. 6. Comparison of indexing algorithms using (a) the value of n as 1 and using (b) the value of n as 3



(a) (b)

Fig. 7. Comparison of indexing algorithms in (a) DB2_A and (b) DB3_A database

4 Conclusions

In this paper, we have briefly and concisely described the main indexing algorithms currently available. Further, they have been classified according to some common characteristics in the methods of index extraction. We have also made an analysis of the advantages and disadvantages of the analyzed algorithms, based on the reliability of the extracted features and the recovery process. Finally a comparison between the exposed algorithms was made using the results reported by the authors in the FVC 2002 database. In Table 2 we can see a summary of advantages and disadvantages of the principal algorithms.

It is important to note that other processes have influenced in the accuracy of indexing algorithms. Some of these processes are: features extraction, enhancement of the fingerprints and the location of the centers of the fingerprints. This occurs because the indexing algorithms make use of features obtained from the previous stages. If these features are not reliable, the indexing algorithms can be seriously affected.

As we can see in Table 2, the bigger efforts of triplet based algorithms are focus on the selection of the triplets and the extracted features. Also, the implementation of filter-based methods is expensive and depends on the location of singular points. We have seen a better performance in algorithms such as Feng *et al.* [6] and Liang *et al.* [9]. In general, we can conclude that algorithms that use features based on triplets of minutiae and ridges have the best performances.

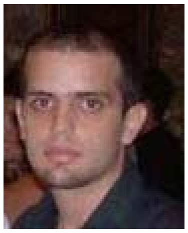
Future work may be directed to new approaches that allow the enrichment of the computed triangulations. Thus, we could achieve better accuracy in cases where some minutiae are not detected.

Table 2. Summary of advantages and disadvantages

Algorithm	Advantages	Disadvantages
Bhanu and Tan [2]	- Robust triplet based features. - Use of geometric constrains.	- All possible triplets are considered.
Germain *et al.* [5]	- Robust triplet based features.	
Bebis *et al.* [1]	- Use of Delaunay triangulations.	- Noise sensitive triplet based features. - Very few features extracted - Poor accuracy.
Liang *et al.* [10]	- Use of higher order Delaunay triangulations. - High accuracy	- Few features extracted
Feng and Cai [6]	- The features extracted from ridges are robust. - High accuracy	- The constructed substructures can be affected by low quality images.
Kumar [7]	- Can be applied to any type of image.	- The implementation is expensive. - Depend on the location of singular points. - Poor accuracy.
Li *et al.* [8]		
Shuai *et al.*	- The number of characteristic points can be regulated.	- Poor characteristic information. - Poor selection strategy of relevant characteristic points.
Boer *et al.* [4]	- The combination of candidates list is an interesting idea.	- Bad implementation of used algorithms.

References

1. **Bebis, G., Deaconu, T. & Georgiopoulos, M. (1999).** Fingerprint Identification Using Delaunay Triangulation. *International Conference on Information Intelligence and Systems*, Bethesda, USA, 452–459.
2. **Bhanu, B. & Tan, X. (2001).** A Triplet Based Approach for Indexing of Fingerprint Database for Identification. *3rd International Conference Audio and Video Based Biometric Person Authentication (AVBPA 2001)*, Halmstad, Sweden. 205–210.
3. **Biswas, S., Ratha, N. K., Aggarwal, G. & Connell, J. (2008).** Exploring Ridge Curvature for Fingerprint Indexing. *2nd IEEE International Conference on Biometrics: Theory, Applications and Systems,* Arlington, Virginia, 1-6.
4. **de Boer, J., Bazen, A. M. & Gerez, S. H. (2001).** Indexing Fingerprint Database Based on Multiple Features. *ProRISC the 12th Annual Workshop on Circuits, Systems and Signal Processing*, Veldhoven, The Netherlands, 300-306.
5. **Germain, R. S., Califano, A. & Colville, S. (1997).** Fingerprint Matching Using Transformation Parameter Clustering. *IEEE Computational Science & Engineering*, 4(4), 42–49.
6. **Feng, J. & Cai, A. (2006).** Fingerprint Indexing Using Ridge Invariants. *18th International Conference on Pattern Recognition (ICPR 2006)*, Hong Kong, China, 4, 433-436.
7. **Singh, J. K. (2009).** *A Clustering and Indexing Technique suitable for Biometric Databases.* MSc Thesis, Indian Institute Of Technology Kanpur, Kanpur, India.
8. **Li, J., Yau, W. & Wang, H. (2006).** Fingerprint Indexing Based on Symmetrical Measurement. *18th International Conference on Pattern Recognition (ICPR'06)*, Hong Kong, China, 1, 1038-1041.
9. **Liang, X., Asano, T. & Bishnu, A. (2006).** Distorted Fingerprint Indexing Using Minutiae Detail and Delaunay Triangle. *3rd International Symposium on Voronoi Diagrams in Science and Engineering (ISVD'06)*, Alberta, Canada, 217–223.
10. **Liang, X., Bishnu, A. & Asano, T. (2007).** 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.
11. **Liu, T., Zhu, G., Zhang, C. & Hao, P. (2005).** Fingerprint Indexing Based on Singular Point Correlation. *IEEE International Conference on Image Processing (ICIP 2005),* Genova, Italy, 3, 293-296.
12. **Mukherjee, R. (2007).** *Indexing Techniques for Fingerprint and Iris Databases.* MSc Thesis, West Virginia University, Virginia, USA.
13. **Shuai, X., Zhang, C. & Hao, P. (2008).** Fingerprint Indexing Based on Composite Set of Reduced SIFT Features. *19th International Conference on Pattern Recognition,* Florida, USA, 1-4.



Alfredo Muñoz Briseño

He works for the Advanced Technologies Application Center (CENATAV) from 2010 as member of the Data Mining's Department. His research interests are focused on fingerprint and geometric graph indexing.

Andrés Gago Alonso

He works for the Advanced Technologies Application Center (CENATAV) from 2006, place where he is a Researcher and the Head of the Data Mining's Department. In CENATAV, his research interests are focused on Data Mining in graph-based content.

José Hernández Palancar

He works for the Advanced Technologies Application Center (CENATAV) from 2003, place where he is a Senior Researcher and Deputy Director for Applied Research, before he was the Head of Data Mining's Department. In CENATAV his research interests focus on Parallel Processing applied to Data Mining and Pattern Recognition algorithms and Biometric.