Using Inverted Index for Fingerprint Search

Johnny Marcos S. Soares[1], Luciano Barbosa[2],
Paulo Antonio Leal Rego[3], Regis Pires Magalhães[1], Jose Antônio F. de Macêdo[3]

[1] Universidade Federal do Ceará, Quixadá, Brazil
johnnymarcos@alu.ufc.br, regis@insightlab.ufc.br
[2] Centro de Informática – CIn, Universidade Federal de Pernambuco, Recife, Brazil
luciano@cin.ufpe.br
[3] Departamento de Computação, Universidade Federal do Ceará, Fortaleza, Brazil
paulo@dc.ufc.br, jose.macedo@insightlab.ufc.br

Abstract. Fingerprints are the most used biometric information for identifying people. With the increase in fingerprint data, indexing techniques are essential to perform an efficient search. In this work, we devise a solution that applies traditional inverted index, widely used in textual information retrieval, for fingerprint search. For that, it first converts fingerprints to text documents using techniques, such as Minutia Cylinder-Code and Locality-Sensitive Hashing, and then indexes them in inverted files. In the experimental evaluation, our approach obtained 0.42% of error rate with 10% of penetration rate in the FVC2002 DB1a data set, surpassing some established methods.

Categories and Subject Descriptors: H.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; I.7 [**Document and Text Processing**]: Miscellaneous

Keywords: Fingerprint indexing, Document retrieval, Inverted index

1. INTRODUCTION

Fingerprints are one of the most used biometric information in verification systems and people identification due to immutability, easy acquisition, and processing speed. In addition, fingerprints have distinguishable characteristics, being unique even in identical twins [Maltoni et al. 2009]. Matching algorithms compare fingerprints using various techniques [Komarinski 2005]. These algorithms use information extracted from fingerprints to compare and generate a score of similarity between them. A significant challenge in performing such a task is that comparing an individual's fingerprint with thousands or millions of other fingerprints is computationally expensive. Many systems use fingerprint indexing techniques to reduce the search space for matching comparisons. In some cases, indexing systems group fingerprints by information such as a fingerprint of a specific finger, which hand has that finger, gender of the person who has a fingerprint, type of fingerprint (based on Henry Classification System [Henry 1900]), among others [Maltoni et al. 2009].

However, due to the population growth and the consequent increase in the number of fingerprints usually enrolled in automatic fingerprint identification systems, it became necessary to create more robust indexing approaches. The characteristics, called minutiae, are the most used information in fingerprint indexing techniques. According to the ANSI / NIST-ITL 2011 standard, minutiae have the following information: position, angle, quality, and type [Mangold 2016].

Previous approaches have used minutiae information for indexing. [Cappelli et al. 2010b] developed a binary representation of the minutiae for indexing, whereas [Khodadoust and Khodadoust 2017]

created triangles from the minutia's position to index them.

Inverted index is a simple but effective indexing strategy and has been widely used for text retrieval. In this work, we investigate the use of inverted indices to index fingerprints based on minutia. This allows the use of existing high-scalable full-text search engines such as Elasticsearch or Solr for fingerprint search. More specifically, we use the Minutia Cylinder-Code (MCC) technique [Cappelli et al. 2010b] that transforms a minutia into a binary vector. Then, Locality Sensitive Hashing (LSH) [Datar et al. 2004] is used to generate terms from n-bits in the vector. To represent the fingerprints we implement two different strategies. The first one creates a single document with n-bit words from all minutiae, while the second approach treats each minutia individually to create sub-documents, which are searched independently. The n-bit terms in the documents are then indexed in an inverted index for searching.

The rest of the article is structured as follows: Section 2 presents concepts necessary to understand the work. Section 3 presents some works related to the proposed method. In Section 4, we explain the indexing and search proposals. Section 5 shows the experiments and results of the proposed methods. In Section 6, we discuss the final considerations of the work and the proposed approaches.

2. THEORETICAL FOUNDATION

In this section, we cover the fundamental concepts for understanding and developing this work.

2.1 Fingerprint

Papillae are structures on the skin that form fingerprints. The papillae are elevations in the skin at the fingertips in the form of lines. These formations vary between people and generally do not change over time. For this reason, Fingerprints have been used since ancient times to identify people and authenticate documents [JUNIOR 1991]. Fingerprints are extracted from the fingers by fingerprint scanners that can be optical, capacitive, ultrasonic, or thermal [Maltoni et al. 2009]. Figure 1 shows examples of fingerprints.

Fig. 1. Examples of fingerprints.

2.2 Fingerprint Features

Fingerprint images have a large amount of information that algorithms can extract. The indexing and matching methods use characteristics extracted to perform the recognition of users. There are three groups of fingerprint features: Level 1, Level 2, and Level 3.

2.2.1 *Level 1 (L1).* Level 1 features are global information from fingerprints. For example, singular points are structures known as the core and delta. The core is a formation located in the central region of the fingerprints and can have a direction. Delta is an expressive angulation of some lines or a formation in the shape of a triangle that can sometimes occur in a fingerprint. Figure 2 shows a core and delta on a fingerprint.

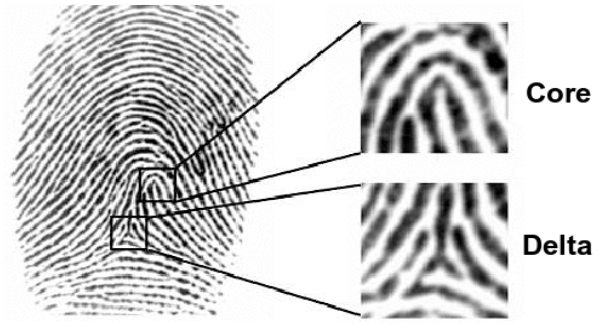

Fig. 2. Singular points on a fingerprint.

2.2.2 *Level 2 (L2).* Also known as local features, L2 features are made of information found in structures called ridges [Maltoni et al. 2009]. The features found in the ridges are called minutiae and have four types. However, the ridge ending and bifurcation are the most recurring features. The ridge ending lines are abruptly ended, and bifurcation lines are divided into two other lines. The minutiae are the fingerprints features most used in matching and indexing algorithms due to the fast processing speed and stability in the quality of results [Holder et al. 2011]. Figure 3 shows the two types of minutiae in a fingerprint.

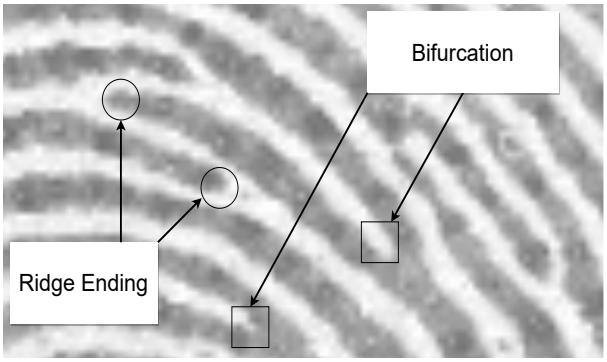

Fig. 3. Bifurcations and ridge endings minutiae on a fingerprint.

2.2.3 *Level 3 (L3).* They are the most specific features among the characteristics described. Features L3 consider the ridges' details, for example, line thickness, shape, contour, curvature, and holes in the lines. However, it takes high-quality images to find this kind of feature.

2.3 Minutia Cylinder-Code (MCC)

Cappelli et al. [2010b] proposed a representation associated with the minutiae of fingerprints. The MCC uses minutiae and information from its neighborhoods to create a cylinder-shaped representation for each minutia. Each cylinder is centered in a minutia m and divided into sections. Each section receives the neighborhood minutiae's spatial and directional contribution with a specific angular difference, compared to the minutiae angle m. A function f generates contributions from the neighborhood minutiae and to those stored in the cylinder sections. Then, Cappelli et al. [2010b] use a threshold δ to binarize the sections and finally organize the sections in a binary vector format. Figure 4 shows the sections of the cylinders with some white regions with contributions from the minutiae of the neighborhood of the minutia m.

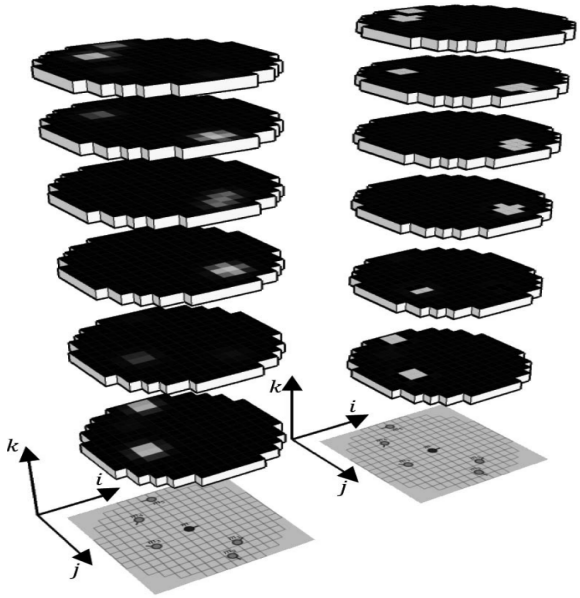

Fig. 4. Two-cylinder sections of the MCC.

3. RELATED WORK

In this section, we present some related work to our proposed solution. First, we discuss about Cappelli et al. [2010a], which proposes the use of MCC and LSH, on which we base ourselves to develop the present work. Next, we show the work of Soares et al. [2020] that use Elasticsearch to perform indexing and search for fingerprints, in which this work is an extension. Paulino et al. [2013] use an approach of joining several indexing methods to generate a similarity score. Finally, we present the work of Moia and Henriques [2018], which use an approach with the MCC and a data structure called Hierarchical Bloom Filter Tree (HBFT).

3.1 Fingerprint Indexing Based on Minutia Cylinder-Code

Cappelli et al. [2010a] use MCC to create a binary representation of each fingerprint's minutiae. For that, they use Locality Sensitive Hashing (LSH) to group segments of the binary vectors of each m_i minutia and store the i information in buckets in hash tables from the entire value generated by the vector segment. L hash tables and l hash functions are used, in which each hash function f_{H_k} in the

range of $f_{H_1}, ..., f_{H_l}$ receives a subvector of size h bits and returns the position of the bucket relative to the decimal number generated by the h bits.

Next, the minutia identifier is inserted into the informed bucket. In the search step, binary vectors are searched, and each hash function will generate a decimal value for a portion of each binary vector. For each decimal value found, a hash table position is accessed to retrieve a list of minutiae. Then, minutiae with a maximum occurrence value are considered similar. All minutiae are queried in the hash tables to select minutiae similar to each minutia searched. Finally, their method creates a ranking of the most similar fingerprints using a distance-based Hamming similarity function [Gionis et al. 1999].

3.2 Indexing Fingerprints Using Inverted Index: An Initial Investigation

The method proposed in Soares et al. [2020] use Elasticsearch as a search engine for documents created from the MCC binary vectors. It uses Locality Sensitive Hashing (LSH) to group segments of the vectors in similar groups. Then, it creates terms that associate the term's position in the vector and the value generated by LSH to form a document with all the terms that will be indexed. Finally, the search is performed using a full-text search with a document of terms. The work proposed in this article is an extension of Soares et al. [2020] work and will refer to it as Approach 1.

3.3 Latent Fingerprint Indexing: Fusion of Level 1 and Level 2 Features

Paulino et al. [2013] propose the use of features Level 1 and Level 2 in indexing fingerprints. This method uses a combination of minutiae, using the technique proposed in Cappelli et al. [2010a], combined with singular points, orientation field, and other general information of the fingerprints. Each indexing method generates a similarity score from comparing a searched fingerprint and each of the indexed fingerprints. The scores' values are then combined to create a final score to rank the most similar fingerprints.

3.4 A New Similarity Digest Search Strategy Applied to Minutia Cylinder-Codes for Fingerprint Identification

Moia and Henriques [2018] propose an approximate fingerprint identification strategy called MCC-HBFT. Their method also uses MCC representation and a data structure called Hierarchical Bloom Filter Tree (HBFT). HBTF is a location-based and probabilistic structure used to store a large amount of data using hash functions and in conjunction with a particular type of tree. Each hash function takes a subvector and indexes it in the HBTF tree.

Each hash function receives a subvector of each of the MCC's binary vectors to perform the searches. Then, searches are made in each of the HBFT to return the elements with the most significant similarity. The following step performs the count of hashes that match the search and creates a ranking based on the counting's ordering.

4. PROPOSED SOLUTION

We propose two different strategies for fingerprint search. The first one creates a text document with all the terms generated by the minutiae, and the second approach builds a sub-document for each minutiae. We implement these approaches in a system composed of the following components, as depicted in Figure 5:

—Fingerprint processing, responsible for processing the fingerprint images and transforming them into textual documents;

—Fingerprint indexing, which processes the documents generated by the previous step and indexes them in a full-text search engine;

—Fingerprint search responsible to search for fingerprints in the index.

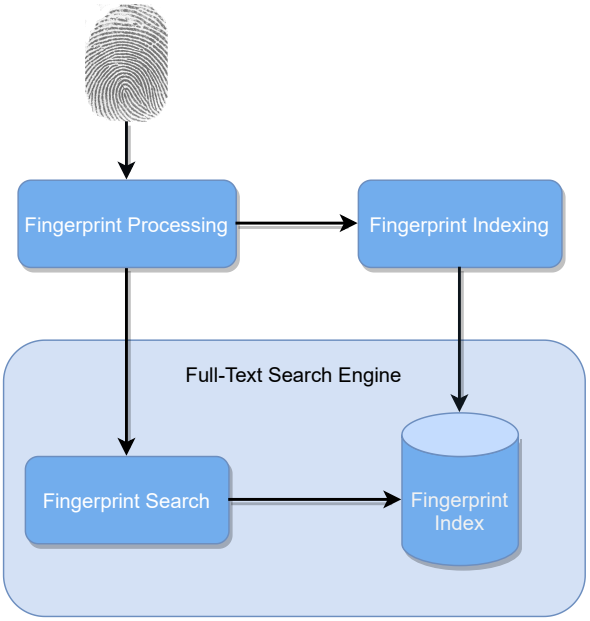

Fig. 5. Architecture of the proposed system to index and search for fingerprints.

4.1 Fingerprint Processing

This module performs the following steps to transform a fingerprint image into a document:

(1) Given the fingerprint image, we use a minutiae extrator to extract the minutiae of fingerprints and their position, angle, and quality information. Figure 6 shows the format of the file containing a minutia per line generated by the MINDTCT tool from the NBIS package (NIST Biometric Image Software)[Ko 2007].

(2) Then, we use the Minutia Cylinder-Code (MCC) technique to generate the binary vectors from the minutia file. The MCC creates an n-bit binary vector for each minutia found. Figure 7 presents the minutiae representation using MCC.

(3) With the binary vectors created, we apply to each vector a function f_H that receives a subvector of h bits and returns the decimal value referring to those h bits. The function f_H creates a term in the format k_b for each subvector and saves them in a document. The value k is the subvector's h bits position in the binary vector, and b is the decimal value returned by the f_H function. The terms that have the return of the f_H function equal to zero are not added in the document because there are many zeros sequences in the binary vectors, which would increase the number of terms.

Figure 8 presents an example of using this strategy after the creation of the binary vectors. In Figure 8(a) there are 5 binary vectors M_1, ...,M_5 of size 9 bits. Figure 8(b) shows the use of the f_H function, considering the size of the subvector h of 3 bits. Figure 8(c) shows the terms generated from the f_H function return, considering the removal of terms with a value of $b = 0$.

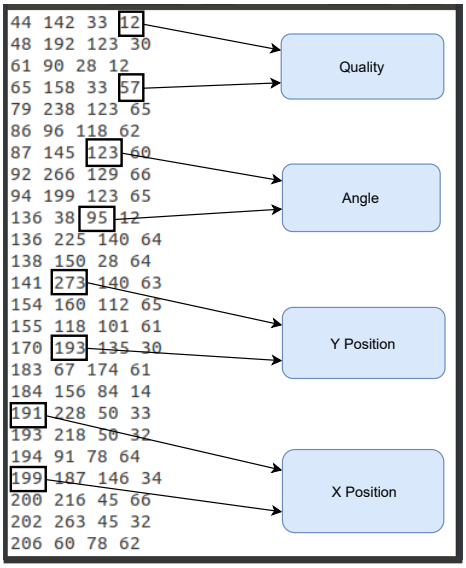

Fig. 6. Example of position, angle, and minutia quality information obtained with the MINDTCT tool.

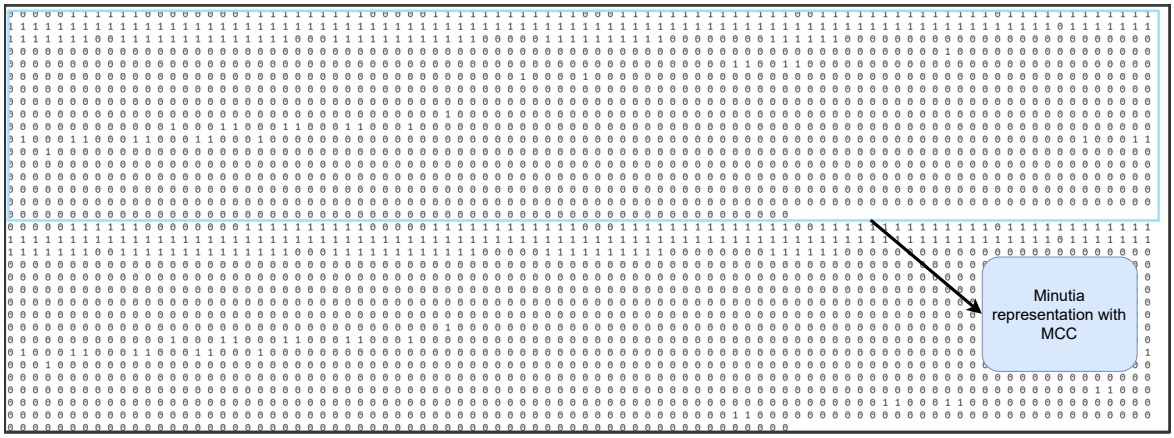

Fig. 7. MCC binary vector representation of minutiae.

4.2 Fingerprint Indexing

This step is responsible for processing the n-bit words that represent a fingerprint's minutia in an inverted file to speed up the search performance.

4.2.1 *Fingerprint–Based Approach.* In this scenario, the n-bit words from all fingerprint's minutiae are indexed as a single document. Figure 9 shows the indexing of the fingerprint data, in which the "minutia" field stores the fingerprint's n-bit vectors. Documents are stored in an inverted index.

4.2.2 *Minutia-Based Approach.* In this strategy, each each fingerprint's minutia is considered a sub-document that composes a document (the fingerprint), as shown in Figure 10. This allows to individually compute similarity for these sub-documents as opposed to the document-based approach.

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9
M_1	0	0	0	1	1	1	0	1	1
M_2	1	0	1	1	1	0	0	0	1
M_3	0	1	1	1	0	1	1	1	0
M_4	1	0	1	1	0	0	0	1	1
M_5	1	1	1	0	1	1	1	0	1

(a) Minutia Cylinder-Code Binary Vectors

f_H	f_H	f_H
0	7	3
5	6	1
3	5	6
5	4	3
7	3	5

(b) Result of the f_H function

2_7 3_3
1_5 2_6 3_1
1_3 2_5 3_6
1_5 2_4 3_3
1_7 2_3 3_5

(c) Generated terms document

Fig. 8. Creation of the terms document from the binary vectors.

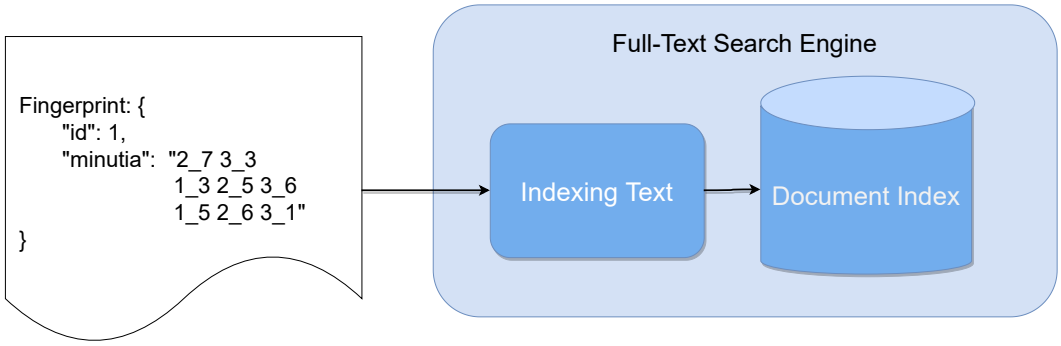

Fig. 9. Example of indexing fingerprint data to the document index in Approach 1.

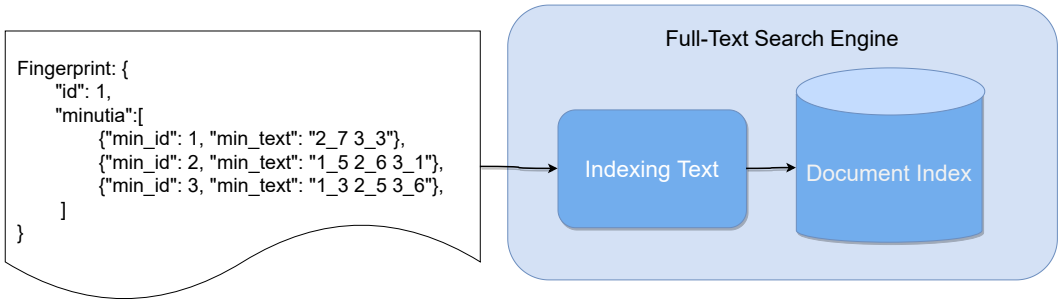

Fig. 10. Example of indexing fingerprint data to the document index in Approach 2.

4.3 Fingerprint Search

Given the fingerprint image that you want to perform a search on the index, our system initially processes it using the Fingerprint Processing module, transforming it into an n-bit vector.

4.3.1 *Fingerprint-Basead Approach.* The search query in this scenario is a document containing the n-bit vector representation of a fingerprint. To retrieve the most similar fingerprints in the index. we use the cosine similarity with td-idf weighting scheme.

4.3.2 *Minutia-Based Approach.* This method performs a search in the index considering each minutia of the queried fingerprint fp_{search} represented by its n-bit vector. More specifically, it calculates the pairwise word overlap between the minutae of fp_{search} and the ones of the fingerprints in the index (fp_{index}). Next, for each minutia of fp_{search} it selects the indexed finprint(s) with the highest overlap, and then sums the scores of all them producing a fingerprint ranking. Figure 11 shows an example of how the ranking is calculated.

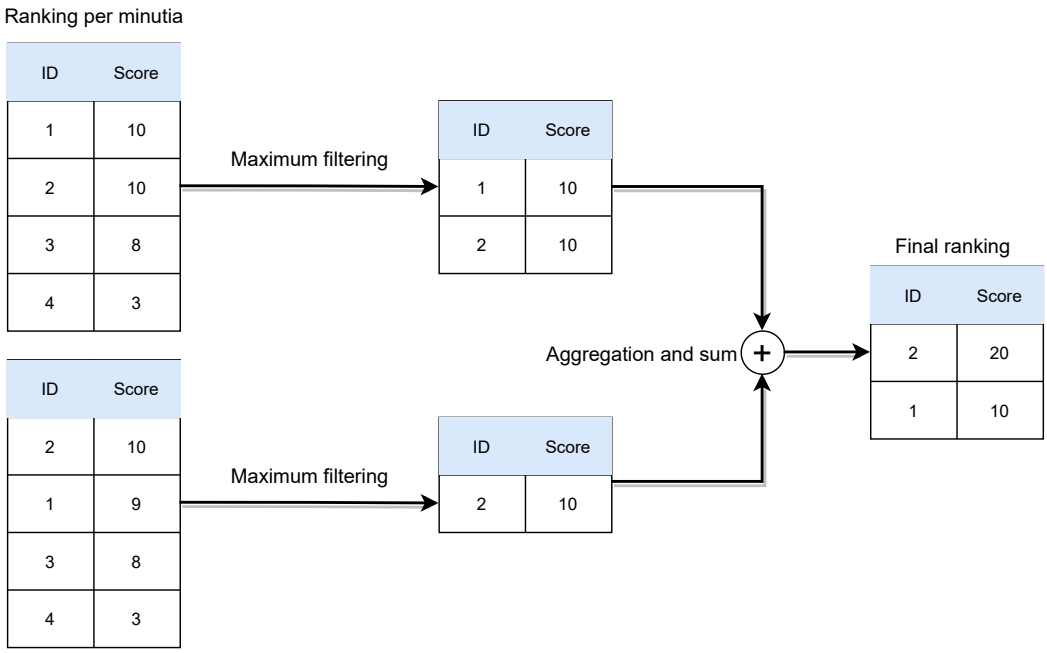

Fig. 11. Example of ranking calculation.

5. EXPERIMENTS AND RESULTS

This section presents the adopted evaluation protocol and describes and discusses achieved results.

5.1 Material and Method

The approaches used in this article proposes the use of the scalability and distribution of the Elasticsearch to perform indexing and searching for fingerprints. Elasticsearch is a search engine that uses an inverted index to perform indexing and searching and can reach petabytes of indexed data [Gormley and Tong 2015]. Also, it has parameters to configure an index in a simple way. For example, the shards parameter partitions the index through the nodes in the cluster. Another critical parameter is the replicas used in the index. They are copies of shard data saved on different nodes, ensuring more data availability. Elasticsearch supports full document queries, called full-text search. This type of query uses similarity models to associate the indexed documents with the searched document. Elasticsearch has several traditional Information Retrieval [Baeza-Yates et al. 1999] models available such as BM25 and cosine similarity.

NIST Biometric Image Software (NBIS) is a public domain software package for extracting and using biometric information. NBIS was developed by the National Institute of Standards and Technology (NIST) for the Federal Bureau of Investigation (FBI) and the Department of Homeland Security

(DHS) for use in processing fingerprint images [Ko 2007]. This work uses the MINDTCT tool from the NBIS package to extract minutiae from a fingerprint image.

5.2 Experiments

Experiments were carried out with a private data set containing 11 thousand fingerprints, in which 10 thousand are used in indexing and 1 thousand in search. This data set's fingerprints are grouped from the value of the quality of the fingerprint, which can be from 0 to 100 using the NFIQ 2 [Bausinger and Tabassi 2011] tool. For these experiments, we have discretized these values in four intervals: 0-25, 26-50, 51-75, and 76-100.

The evaluation of fingerprint indexing methods is usually carried out with the relationship between Penetration Rate and Error Rate. Penetration Rate is the percentage of the base that needs to be researched to find the correct result. The Error Rate is the percentage of searches that did not obtain the correct result within the Penetration Rate limit.

Figure 12 shows the graph of Penetration Rate X Error Rate from Approach 1 in the private data set, with curves referring to quality groups of the search fingerprints. We can observe that the higher the image quality, the better the search performance. The curve for the highest quality fingerprints (quality values between 76 and 100) has a Error Rate of 0%, that is, all the fingerprints searched for were found. Considering the objective of dealing with scalability and, consequently, providing efficient searches, another important factor in analyzing the solution is the response time. The average time per query in these experiments in Approach 1 was 0.2 seconds, which confirms the search approach's efficiency. However, it was not possible to execute Approach 2 on the dataset since Approach 2 did not scale to large volumes of data. Besides, it was not possible to verify the search time for other methods in the data set used, as the source codes were not published.

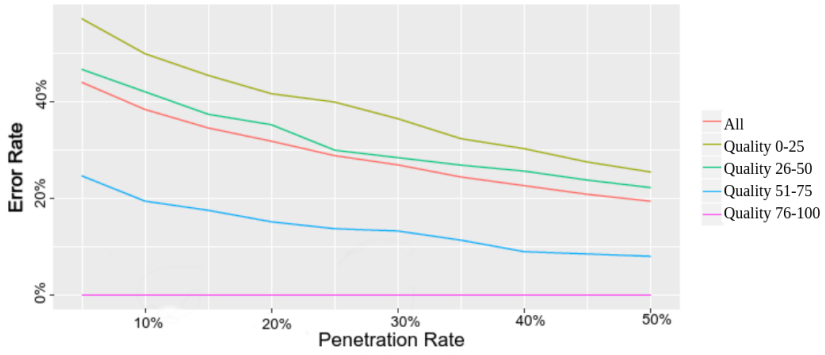

Fig. 12. Performance chart of Approach 1 on the private dataset.

Another public dataset was used (FVC2002 DB1a [Maio et al. 2002]) for comparison with the fingerprint indexing technique proposed by [Cappelli et al. 2010a], as they used this base for evaluation. The data set has 800 fingerprint images from 100 fingers, totaling 8 images per finger. One fingerprint of each finger is considered the main one and will be indexed in Elasticsearch, while the other 7 are used to perform the searches.

Approach 1 obtained 8.7% Error Rate with 10% Penetration Rate in FVC2002 DB1a. The method proposed in Cappelli et al. [2010a] obtained 1% Error Rate with 10% Penetration Rate in FVC2002 DB1a. Approach 2 obtained 0.42% Error Rate with 10% Penetration Rate. Table I shows the parameters that obtained the best results in the experiments of the approaches proposed in this work using public data. Figure 13 shows the results of the work of Kavati et al. [2017], Khodadoust and

Khodadoust [2017], Cappelli et al. [2010a], Feng and Cai [2006] and the two approaches of this work on FVC2002 DB1a.

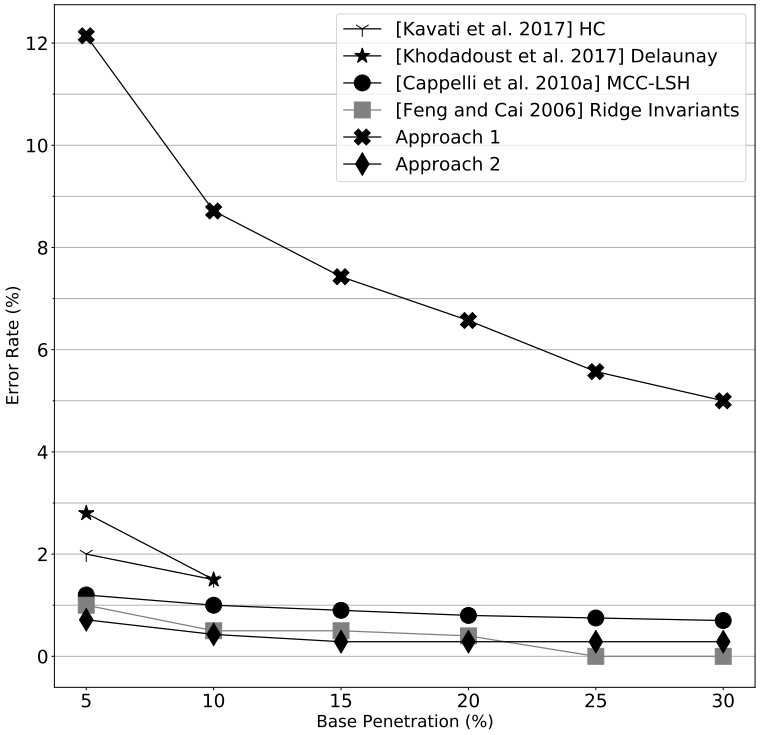

Fig. 13. Performance chart and method comparison.

Parameter	Description	Approach 1 in FVC2002 DB1a	Approach 2 in FVC2002 DB1a
t	Number of terms used in MLT	250	NA
n	Binary vector size	1944	1944
h	Number of bits used in f_H	8	8
T_q	Minutiae quality threshold	15	NA
M_q	Maximum number of minutiae	NA	100

Table I. Values of the best parameters used in the experiments.

Approach 1 obtained, therefore, a lower result than the compared works. However, the fingerprint indexing methods that use minutiae perform individual searches for each minutia in the data set. Nevertheless, Approach 1 performs a single search using all minutiae. This generalization of minutiae decreases the quality of the result. However, it becomes faster in environments with a large amount of data. Approach 2 surpassed the baselines considered, in terms of Error rates, for Penetration Rates between 5% and 10% of the base, as shown in Figure 13. However, as several searches are performed for a fingerprint, the technique has scalability problems. Approach 2 can use Elasticsearch's indexing facility to index a large set of fingerprint data. However, the search time is extremely high, as a search is performed for each minutia. The use of multiple searches on Elasticsearch is not optimized for performing many searches on the same request.

In addition, the proposed approaches are sensitive to the use of low-quality minutiae. For example, in experiments with a low-quality threshold in Approach 1 and with the maximum number of minutiae

high in Approach 2, the methods obtained the worst results. Another critical factor is the quality of the fingerprints, as both approaches were wrong with lower quality fingerprints.

6. CONCLUSION

In this work, we propose to index fingerprints based on Minutia Cylinder-Code and the use of Elasticsearch. The approaches use methods employed in textual searches in the domain of fingerprints. Therefore, using Approach 1, it was possible to index a large number of documents generated from fingerprints and take advantage of the easy scalability and distribution of Elasticsearch. However, indexing at the level of minutia employed in Approach 2 did not escalate to larger data sets, as many requests are made to Elasticsearch to address each minutia individually. Therefore, for future work, document indexing needs to be a middle ground between the two approaches, considering a minutial approach like Approach 2 to get better results, along with the use of a single search similar to Approach 1 to scale for large datasets.

Acknowledgment. The authors would like to thank the Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico (FUNCAP) for the financial support (Process 8789771/2017).

REFERENCES

Baeza-Yates, R., Ribeiro-Neto, B., et al. *Modern information retrieval*. Vol. 463. ACM press New York, 1999.

Bausinger, O. and Tabassi, E. Fingerprint sample quality metric nfiq 2.0. *BIOSIG 2011–Proceedings of the Biometrics Special Interest Group*, 2011.

Cappelli, R., Ferrara, M., and Maltoni, D. Fingerprint indexing based on minutia cylinder-code. *IEEE transactions on pattern analysis and machine intelligence* 33 (5): 1051–1057, 2010a.

Cappelli, R., Ferrara, M., and Maltoni, D. Minutia cylinder-code: A new representation and matching technique for fingerprint recognition. *IEEE transactions on pattern analysis and machine intelligence* 32 (12): 2128–2141, 2010b.

Datar, M., Immorlica, N., Indyk, P., and Mirrokni, V. S. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry*. pp. 253–262, 2004.

Feng, J. and Cai, A. Fingerprint indexing using ridge invariants. In *18th International Conference on Pattern Recognition (ICPR'06)*. Vol. 4. IEEE, pp. 433–436, 2006.

Gionis, A., Indyk, P., Motwani, R., et al. Similarity search in high dimensions via hashing. In *International Conference Very Large Data Bases*. Vol. 99. pp. 518–529, 1999.

Gormley, C. and Tong, Z. *Elasticsearch: the definitive guide: a distributed real-time search and analytics engine*. O'Reilly Media, Inc., 2015.

Henry, E. Classification and uses of fingerprints london. *George Rutledge and Sons, Limited* vol. 54, 1900.

Holder, E. H., Robinson, L. O., and Laub, J. H. *The Fingerprint Sourcebook*. US Department. of Justice, Office of Justice Programs, National Institute of Justice, 2011.

JUNIOR, G. A papiloscopia nos locais de crime: Dactiloscopia, quiroscopia, podoscopia. *São Paulo: Editora Ícone*, 1991.

Kavati, I., Prasad, M. V., and Bhagvati, C. Hierarchical decomposition of extended triangulation for fingerprint indexing. In *Efficient Biometric Indexing and Retrieval Techniques for Large-Scale Systems*. Springer, pp. 21–40, 2017.

Khodadoust, J. and Khodadoust, A. M. Fingerprint indexing based on expanded delaunay triangulation. *Expert Systems with Applications* vol. 81, pp. 251–267, 2017.

Ko, K. User's guide to nist biometric image software (nbis). Tech. rep., 2007.

Komarinski, P. *Automated fingerprint identification systems (AFIS)*. Elsevier, 2005.

Lin, J. and Dyer, C. Data-intensive text processing with mapreduce. *Synthesis Lectures on Human Language Technologies* 3 (1): 1–177, 2010.

Maio, D., Maltoni, D., Cappelli, R., Wayman, J. L., and Jain, A. K. Fvc2002: Second fingerprint verification competition. In *Object recognition supported by user interaction for service robots*. Vol. 3. IEEE, pp. 811–814, 2002.

Maltoni, D., Maio, D., Jain, A. K., and Prabhakar, S. *Handbook of fingerprint recognition*. Springer Science & Business Media, 2009.

Mangold, K. C. Data format for the interchange of fingerprint, facial & other biometric information ansi/nist-itl 1-2011 nist special publication 500-290 edition 3. Tech. rep., 2016.

Moia, V. H. G. and Henriques, M. A. A. A new similarity digest search strategy applied to minutia cylinder-codes for fingerprint identification. In *Anais Principais do XVIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*. pp. 99–112, 2018.

Paulino, A. A., Liu, E., Cao, K., and Jain, A. K. Latent fingerprint indexing: Fusion of level 1 and level 2 features. In *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*. IEEE, pp. 1–8, 2013.

Soares, J. M. S., Barbosa, L., Rego, P. A. L., Magalhaes, R. P., and de Macêdo, J. A. F. Indexando impressões digitais utilizando índice invertido: Uma investigaç ao inicial. *Simpósio Brasileiro de Banco de Dados (SBBD)*, 2020.