

Fast Fingerprint Recognition Using Circular String Pattern Matching Techniques

Oluwole Ajala, Moudhi Aljamea, Mai Alzamel, Costas S. Iliopoulos
Yoann Strigini

Department of Informatics
King's College London
London, UK

e-mail: {oluwole.ajala, mudhi.aljamea, mai.alzamel, costas.iliopoulos, yoann.strigini}@kcl.ac.uk

Abstract—The performance of Automated Fingerprint Identification System (AFIS) heavily relies on how efficiently minutiae are extracted. Most, if not all, AFIS compare minutiae information (such as ridge endings and bifurcation position) in form of sets of coordinates for verification or identification. Surprisingly, research on alternative minutiae extraction schemes is scarce. This paper, proposes the implementation of the novel approach to fingerprint recognition based on the extraction of minutiae in form of circular strings, which are suitable for approximate circular string matching. In addition to that, the proposed solution is able to detect the exact location and rotation of the input fingerprint regardless of its location on the scan surface.

Keywords—Biometrics; Fingerprints; Matching; Verification; Orientation Field.

I. INTRODUCTION

Previously, a plethora of schemes for identification such as knowledge based schemes like passwords, Personal Identification Number (PIN) and token based schemes like passports, driving license were used for identification purposes. However, with the emergence of the internet, the need for automatic person identification has become imperative. Specially with the increase of the dynamic nature of personal activities, particularly business and industry. Consequently, biometric means of attaining this has become predominant. [1], [2].

Biometrics has to do with the metrics or statistical analysis of biological data which can be human traits or characteristics. Biometric identifiers are peculiar and unique to individuals; personal identification based on biometric data offer the most accurate means of identification, hence, among all other forms of biometrics such as eye, face, voice and speech [3], the fingerprint identification remains the most popular till date. Fingerprints have provided an impeccable means of user authentication and personal identification for a long time, possibly dating back to the 19th century, when the records of fingerprint details of criminals in Argentina were released [4]. It has long since been adopted not just for law enforcement purposes (forensics and police) but also for commercial purposes like financial transactions and most recently, it is used as an authentication method in mobile devices and computers. With regards to application, two kinds of fingerprint recognition systems exist (identification and

verification). In the identification system, the query fingerprint is inputted and then matched against a computed list of stored fingerprints for resemblance. In this case, the output will be understandably short or non-existent as no two fingerprints are alike. The verification system however, involves an input of query fingerprints with claimed identities, to be matched against already stored IDs (name and fingerprint) within a database to corroborate consistency. The system then outputs a result which can be either an affirmative or a negative message. The bulk of research that has focused on fingerprint authentication, has however, neglected the rotational issues that arise with fingerprints resulting to incorrect orientation identification. This is because it is assumed and often times wrongly, that the direction of the fingerprint will align with the stored fingerprint image. This singular issue poses tension in fingerprint matching, which only a negligible number in the literature [5] have considered. As computers and mobile devices adopt fingerprint recognition as a way to authenticate user, this apparent tension gains more popularity, becoming an integral research area which must be addressed.

A. Our Contribution

Responding to this rotational issue, this paper proposes a novel pattern matching technique that caters for orientation differences in fingerprints. Despite a plethora of fingerprint matching algorithms, there is still room for improvement [6]. Our proposed solution will employ A Novel Pattern Matching Approach for Fingerprint-based Authentication proposed by [7], by implementing a pre-matching stage called the orientation identification stage and then match the fingerprint image with stored images using an efficient, error tolerant, pattern matching algorithm. The fingerprint is intercepted with a series of scan circles and the minutiae information is derived. This information will then be translated into a string. This fingerprint string information is now matched against a database of stored images using approximate string matching techniques. With this approach, identification of fingerprints can be done in linear time, with respect to the total length of all strings to be searched [7].

B. Road Map

The organization of the rest of this paper is as follows. In Section II, we present some background related to fingerprints. Section III presents a very brief literature review. We present our approach in Section IV. The experiment and the result analysis will be presented in section V. Finally, we briefly conclude and state the future work in VI.

II. BACKGROUND

Fingerprints are made up of minutiae, which are basically ridges and furrows in parallelism with each other. These minutiae form a complicated pattern that when impressed on a fingerprint scanner, leaves a print. These prints are matched to stored images on a database for either verification, authentication or both purposes. The fundamental fingerprint (FP) patterns that exist are whorl, loop and arch [8]. However, a commonly used classification is the Henrys classification [9] [10] consisting of eight classes: Plain Arch, Tented Arch, Left Slant Loop, Right Slant Loop, Plain Whorl, Double Loop Whorl, Central Pocket Loop Whorl and Accidental Whorl (see Figure 1).

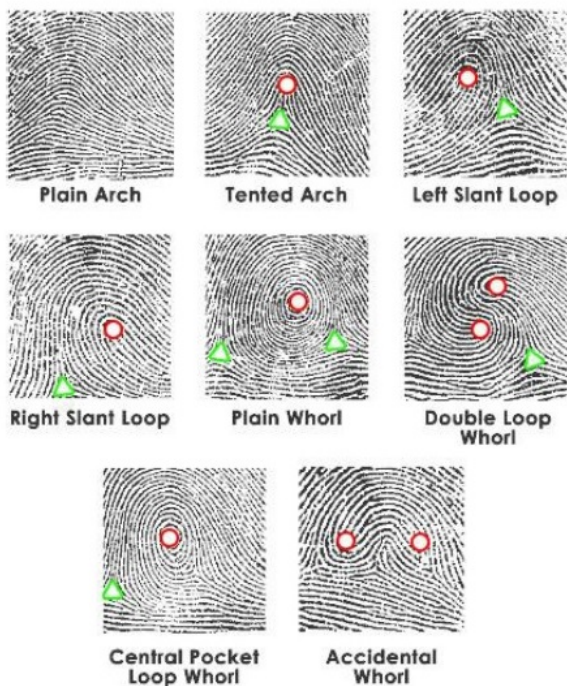


Figure. 1. Classification of fingerprint patterns

Each fingerprint is permanent and of course unique. This distinctiveness is derived by features such as ridges-ridge endings, ridge bifurcation, valleys and furrows referred to as minutiae which form a unique pattern. Recent studies have shown that the probability of two persons sharing same fingerprint is less than one in a billion [11], hence its uniqueness. In early cultures, fingerprints have been relied on to identify

individuals using the so-called ink-technique [12]. This ink-technique required that a persons fingers were first coated in printing ink to get an impression on paper cards. This copy was then scanned to get a digital image. The ink-technique, though an off-line obtainment method, is still applicable today especially in forensic studies as fingerprints often have to be gathered from crime grounds. However, it is impractical for biometric studies [13]. The alternative approach is of course to scan and match fingerprints in real time.

III. RELATED WORKS

Fingerprint recognition has been a core study since prehistoric times, leading to the proposal of several algorithms to developing an almost precise recognition system. Literature on fingerprint recognition has attempted to cover a wide span on the minutiae of fingerprints [14], [15], [16], [17]. Additionally, the memory and processor intensive computation issues has been discussed and addressed in some previous works. However, most of these recognition approaches hinge on the assumptions that the fingerprint impression was got from a vertically placed finger to produce a linear pattern. The minutiae based matching remain the most popular approach. This is because minutiae are believed to be the most discriminating and reliable features [18].

These previous works that have been grounded on the fingerprint minutiae recognition ignored to deliberate on the image distortions that can occur when obtaining a print with different rotation (see Figure 2). As a result, researchers have also used other features for fingerprint matching. For example, the algorithm in [19] works on a sequence of points in the angle-curvature domain, after transforming the fingerprint image into these points. A filter-based algorithm using a bank of Gabor filters to capture both local and global details in a fingerprint as a compact fixed-length finger code is presented in [20]. In the literature, the combination of different kinds of features have also been studied [21], [22]. There exist various other works in the literature proposing different techniques for fingerprint detection based on different feature sets of fingerprints [16], [23], [24]. Due to brevity we do not discuss these works in this paper. Interested readers are referred to a very recent review by Unar et al. [2] and references therein.

Note that, in addition to a large body of scientific literature, a number of commercial and proprietary systems are also in existence. In the related industry, such systems are popularly termed as Automatic Fingerprint Identification System (AFIS). A problem with the Automated Fingerprint Identification system (AFIS) has to do with the sensors used to capture the image. It is impractical to assume that the fingerprints to be compared are obtained from a central sensor. This will inevitably lead to a conflict in pattern matching when a different sensor is used [25].

Also, with the commercially available AFISs, there poses the challenge of increasing the matching speed without compromising the accuracy in the application context of identification, more so, when the database is large [6]. For this reason,

the quest for yet a better fingerprint recognition algorithm is nascent [6].

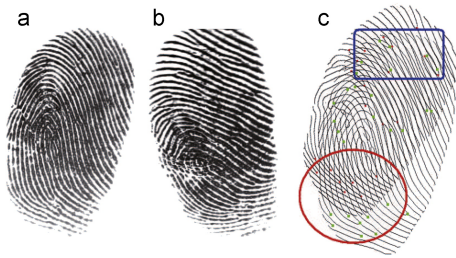


Figure. 2. An example of large distortion from FVC2004 DB1 [26]

IV. OUR APPROACH

As opposed to gathering information about ridge endings and bifurcations from each fingerprint, the proposed algorithm extracts minutiae information in form of circular strings. Thereafter, the Approximate Circular String Matching via Filtering (ACSMF) algorithm [27] is applied to the circular strings, to find all occurrences of the rotations of a pattern of length m in a text of length n [28], where n is the concatenation of all string representations of the fingerprints in the database, and m is the string representation of the fingerprint to identify. It follows that complexity of this approach is $O(n)$. The solution proposed in [7] is divided into two main stages:

- Stage 1 Orientation Identification
- Stage 2 Verification and Matching

A. Algorithmic Overview

```

1  ALGORITHM novel-minutiae-extraction (char[][] img, int r, int cx, int cy)
2  //INPUT: "img", a 2d char array representing a fingerprint scan
3  //INPUT: "r", the radius of the current circular scan
4  //INPUT: (cx, cy) the coordinates of the center of the current circular scan
5  //OUTPUT: "pattern", a circular binary string
6  FOR i FROM cx - r TO cx + r
7      FOR j FROM cy - r TO cy + r
8          IF  $r^2 = (i - cx)^2 + (j - cy)^2$  THEN
9              // pixel img[i][j] is at the intersection of the FP with the circle
10             IF  $x < cx$  THEN
11                 // Left half of scan circle
12                 IF pixel at img[i][j] < 125 THEN
13                     append 0 to the left of pattern
14                 ELSE
15                     append 1 to the left of pattern
16             ELSE
17                 // Right half of scan circle
18                 IF pixel at img[i][j] < 125 THEN
19                     append 0 to the right of pattern
20                 ELSE
21                     append 1 to the right of pattern
22  RETURN pattern
    
```

Figure. 3. Minutiae Extraction Algorithm.

B. Details of Stage 1: Orientation Identification

In this stage, we employ a novel approach based on circular templates as follows. Let us use f_i to denote the image of the input fingerprint. Let us assume that we know the

appropriate center point, p of f_i . We then can convert f_i to a representation consisting of multiple circular bit streams by extracting circular segments of the image as shown in the algorithmic overview /Figure 3. This is achieved by constructing k concentric circles C_j of radius $r_j, 1 \leq j \leq k$, with center at point p . For each circle, we obtain minutiae features of the image by storing 1 wherever the edge of a circle intersects with a ridge and a 0 if it intersects with a furrow. So, in this way, for f_i , we get k concentric circles, which can be transformed into k circular binary strings see Figure 4. Clearly, this procedure can be easily applied on a fingerprint data stored on the database. In what follows, we will use $Y_j, 1 \leq j \leq k$ to denote the k circular strings obtained after applying the above procedure on a fingerprint data stored in the database. In what follows, we may slightly abuse the notation and say the Y_j corresponds to the circle of radius r_j .

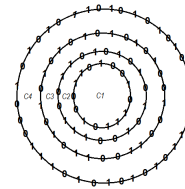


Figure. 4. Intersection of a circle with the fingerprint

Now, to identify the location and orientation of the input fingerprint we generalize the above approach to extract the minutiae feature and apply the approximate circular string matching algorithm of [28] as described in Figure 5. What we do is as follows. For the input fingerprint, we cannot assume a particular center point to draw the concentric circle which is actually the main reason for difficulty in the process. So, instead, we take reference points at regular intervals across rows and columns of the entire frame of the image (i.e., the input scanning area) and at each point p_ℓ , concentric circles $C_{j\ell}$ of radius r_j are constructed. Like before, k is the number of circles at each reference point p_ℓ . So, from the above procedure, for each point p_ℓ we get k circular strings $X_{j\ell}, 1 \leq j \leq k$.

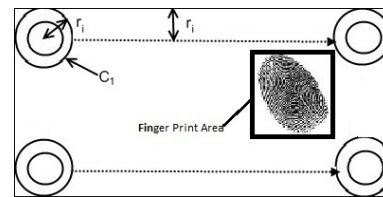


Figure. 5. Identifying the orientation and surface area of the fingerprint impression

At this point the problem comes down to identifying the best match across the set of same radius circles. To do this we make use of the Approximate Circular String Matching via Filtering (ACSMF) algorithm, presented in [28], which is accurate and extremely fast in practice. To do this we take a particular $X_{j\ell}$, construct $X_{j\ell}.X_{j\ell}$ (to ensure that all conjugates of $X_{j\ell}$ are considered) and apply algorithm ACSMF on $X_{j\ell}.X_{j\ell}$ and

Y_j . In other words, we try to match the circular string Y_j (corresponding to the circle of radius r_j) to all circular strings $X_{j\ell}$ (corresponding to the circle of radius r_j) generated at each point p_ℓ . Thus we can identify the best matched circular string, i.e., the best matched circles and thereby locate and identify the fingerprint impression with the correct orientation. Once the orientation has been identified, we can apply standard techniques to reorient the image to match with the image from the database in the next stage.

C. Details of Stage 2: Verification and Matching

As with most other fingerprint recognition systems, a database with fingerprint information is kept. It is against this, that the queried fingerprint will be matched. Once stage 1 (the orientation identification stage) is complete, we can then simply re-orient the fingerprint impression to suit the stored format in the database, then the matching algorithm runs on an assumed dual image of the same orientation and magnitude. This is called the verification and can be effectively carried out thus. Each image, now viewed as a two dimensional matrix, consisting of zero and one values can be converted to a binary string (one dimension). At this point, we are left with just pattern matching between two strings of equivalent length. However, note that the possibility of errors must be considered here. Hence, we simply compute the edit distance between the two binary strings and if the distance is within the tolerance level, we consider the fingerprint to be recognized. Otherwise, the authentication fails. In fact, the used matching algorithm (Levenshtein algorithm) computes the best alignment by using an edit distance which simply states the number of differences that must be changed to attain a perfect match, without considering error possibilities,

V. THE EXPERIMENT

The proposed approach has been developed in ANSI C/C++ using the external library OpenCV (freely available for academic use, under the BSD licence, at <http://opencv.org>) for standard image processing. Different inputs have been tested by running the `fp_auth` several times against the Fingerprint Special Database of the National Institute of Standards (NIST) [29]. All external sources are open/free for academic purposes under (BSD licence). The experiment has been tested with black and white Tiff images. These images have been pre-processed to be thinned fingerprints using C++ implementation of the Guo-Hall image thinning algorithm [30]. The results in (Table I) shows the experiment results over enhanced images with different parameters.

The data entries in the table are explained as follows: Mated image refers to the input image whether it is related to the compared image or not, No. of mismatch allowed is the tolerance threshold under which the input fingerprint is to be considered as candidate match corresponding to the set of circular strings, , Max radius, is the radius for the maximum circle by pixels that can be scanned per image, Radius distance, is the number of interval in pixels between

TABLE I. EXPERIMENT RESULTS 1

Mated Image	No. Mismatch allowed	Max Radius	Radius distance	Elapsed time of get scans	No. Matches	Rotation in pixels
Y	10	60	2	0.7197	0	0
Y	30	50	2	0.7308	1	10
Y	30	60	10	0.077261	6	<10
Y	30	60	5	0.292098	5	<10
Y	50	60	2	0.6865	>30	10
Y	60	60	10	0.074452	>30	<10
N	80	60	2	0.7823	2	137
N	20	60	10	0.054203	0	0

each circle centre point. Elapsed time to get scans is the time in seconds to get the total circular scans per image. No. matches is the number of candidate matches after applying the ACSMF algorithm. Finally, Finally, Rotation in pixels is the rotation to be applied on the input fingerprint image in pixels.

In particular, the table shows that the time to get scans for each image is less than a second. Essentially, it displays that increasing the number of allowed mismatch, will result in increasing the number of matched candidates returned by ACSMF. For instance, when mated images are scanned and compared when the number of allowed mismatch is very low, almost equal to 10, it results to a negative return. . In contrast, when the number of mismatch allowed is very high for example 80, the number of returned matches is 2 even though the input image is different to the stored image. However, the correct match is shown when the number of mismatch is equal to 30. In general, the results show the effect of choosing the number of mismatch allowed which should not be very high to avoid false positive returns nor very low to prevent the false negative rate either. Finally, the results indicates the direct proportion between the circles centre point in each image and the scanning speed. Finally, the results indicate the direct proportion between the circles centre point in each image and the scanning speed.

The main advantage of this approach is regardless of the fingerprint rotation degree, the accuracy of the result will not be affected, whereas most of the other finger print detection algorithms accuracy results are affected by the rotation degree. Moreover, according to the Fingerprint Matching and Non-Matching Analysis for Different Tolerance Rotation Degrees study in [31] where they evaluated three biometric systems Neurotechnology Verifinger 6.0 Extended, Innovetrics IDKit SDK and Griaule Fingerprint SDK 2007 and the influence of the fingerprint rotation degrees on false match rate (FMR), their results showed that the FMR values increase as rotation degrees increase too. Additionally, it was stated that one of the factors that affect the performance of the matching algorithm is the fingerprint rotation. However, this is not the case in our approach.

A. Accuracy and Speed

We have two parameters that determine the accuracy of our approach. In Stage 1, the accuracy depends on the number of concentric circles, k . The larger the value of k , the higher the accuracy of pinpointing the location with the correct orientation. However, as k increases the computational requirement and time also increases. In Stage 2, we have another parameter d which is the tolerance level, i.e., the (edit) distance allowed between the two strings. At this point a brief discussion on the response time of our algorithm is in order. Note that, the bulk of the computational processing in our approach is required in Stage 1, where we apply algorithm ACSMF to identify the best matched circles. As has been shown in [28], on average, ACSMF works in linear time to the size of the input and is extremely fast. The size of the circles and hence the corresponding circular strings are very small and can be assumed to be constant for all practical purposes. As a result the running time of Stage 1 would be extremely fast. Again, since the size of the fingerprint image is very small, any efficient approximate string matching algorithm in Stage 2 would give us a very quick result. As an example, for an image of 300x300 pixels, extracting 45 circular strings having radius of 60 pixels takes on average 0.2200 seconds. Furthermore, the per-column loop nested inside the per-row loop allows for a 10x faster access this is because of the way C accesses memory.

B. Two Modes of Fingerprint Recognition System

As has been mentioned before, in terms of applications, there are two kinds of fingerprint recognition systems. So far, we have only considered the mode where the input is a query fingerprint with an identity (ID) and the system verifies whether the ID is consistent with the fingerprint (i.e., verification mode). Here, the output is an answer of *Yes* or *No* and we need only match against one fingerprint from the database (i.e., the finger print coupled with the ID). To handle the other mode (identification mode), we need to match the query fingerprint against a list of fingerprints in the database. This can be done using an extension of algorithm ACSMF, namely Approximate Circular Dictionary Matching via Filtering algorithm (ACDMF) [32]. We omit the details here due to space constraints. Both ACSMF and ACDMF implementations are available at [27].

VI. CONCLUSION AND FUTURE WORK

This paper proposes yet a new pattern matching based approach for fast and accurate recognition of fingerprints. A notable challenge in fingerprint matching is that the rotation of the fingerprint is assumed to be in sync with the stored image; in this paper we have tackled this issue. The novel element of this paper is the process of using a series of circles to transform minutiae information into string information consisting of 0s and 1s, and then using the approximate circular string matching algorithm to identify the orientation. This technique has improved the performance and accuracy

of the fingerprint verification system. Although our matching algorithm produces nearly accurate results at high speed, implementing the suffix tree technique to this approach will improve the accuracy and speed for big volume data.

REFERENCES

- [1] S. Sebastian, "Literature survey on automated person identification techniques," *International Journal of Computer Science and Mobile Computing*, vol. 2, no. 5, pp. 232–237, May 2013.
- [2] J. Unara, W. C. Senga, and A. Abbasi, "A review of biometric technology along with trends and prospects," *Pattern Recognition*, vol. 47, no. 8, pp. 2673–2688, August 2014.
- [3] P. Szor, *The art of computer virus research and defense*. Addison-Wesley Professional, 2005.
- [4] National Criminal Justice Reference Service, *The Fingerprint Sourcebook*, A. McRoberts, Ed. CreateSpace Independent Publishing Platform, 2014.
- [5] A. Agarwal, A. K. Sharma, and S. Khandelwal, "Study of rotation oriented fingerprint authentication," *International Journal of Emerging Engineering Research and Technology*, vol. 2, no. 7, pp. 211–214, 2014.
- [6] P. Gutierrez, M. Lastra, F. Herrera, and J. Benitez, "A high performance fingerprint matching system for large databases based on gpu," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 1, pp. 62–71, 2014.
- [7] M. Aljamea, T. Athar, C. S. Iliopoulos, S. P. Pissis, and M. S. Rahman, "A novel pattern matching approach for fingerprint-based authentication," in *PATTERNS 2015: The Seventh International Conferences on Pervasive Patterns and Applications*. IARIA, 2015, pp. 45–49.
- [8] K. H. Q. Zhang and H. Yan, "Fingerprint classification based on extraction and analysis of singularities and pseudoridges," in *Proceedings of the Pan-Sydney area workshop on Visual information processing-Volume 11*, ser. VIP 2001. Sydney, Australia: VIP, 2001, pp. 83–87.
- [9] E. R. Henry, *Classification and Uses of Finger Prints*. Routledge, 1900.
- [10] H. C. Lee, R. Ramotowski, and R. E. Gaensslen, Eds., *Advances in Fingerprint Technology, Second Edition*. CRC Press, 2002.
- [11] S. Sebastian, "Literature survey on automated person identification techniques," *International Journal of Computer Science and Mobile Computing*, vol. 2, no. 5, pp. 232–237, 2013.
- [12] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*. Springer-Verlag, 2009.
- [13] Griaule Biometrics. (2014, Nov.) Online and offline acquisition. [retrieved: 01.2016]. [Online]. Available: <http://www.griaulebiometrics.com/en-us/book/>
- [14] X. Tan and B. Bhanu, "Fingerprint matching by genetic algorithms," *Pattern Recognition*, vol. 39, no. 3, pp. 465–477, 2006.
- [15] A. K. Jain, L. Hong, S. Pankanti, and R. Bolle, "An identity-authentication system using fingerprints," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1365–1388, 1997.
- [16] Z. M. Kovacs-Vajna, "A fingerprint verification system based on triangular matching and dynamic time warping," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 11, pp. 1266–1276, 2000.
- [17] X. Tan and B. Bhanu, "Robust fingerprint identification," in *International Conference on Image Processing 2002*, vol. 1. IEEE, 2002, pp. 1–277.
- [18] C. Kai, Y. Xin, C. Xinjian, Z. Yali, L. Jimin, and T. Jie, "A novel ant colony optimization algorithm for large-distorted fingerprint matching," *Pattern Recognition*, vol. 45, no. 1, pp. 151–161, 2012.
- [19] A. A. Saleh and R. R. Adhami, "Curvature-based matching approach for automatic fingerprint identification," in *System Theory, 2001. Proceedings of the 33rd Southeastern Symposium on*. IEEE, 2001, pp. 171–175.
- [20] A. K. Jain, S. Prabhakar, L. Hong, and S. Pankanti, "Filterbank-based fingerprint matching," *Image Processing, IEEE Transactions on*, vol. 9, no. 5, pp. 846–859, 2000.
- [21] A. Jain, A. Ross, and S. Prabhakar, "Fingerprint matching using minutiae and texture features," in *Image Processing, 2001. Proceedings. 2001 International Conference on*, vol. 3. IEEE, 2001, pp. 282–285.
- [22] A. V. Ceguerra and I. Koprinska, "Integrating local and global features in automatic fingerprint verification," in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, vol. 3. IEEE, 2002, pp. 347–350.

- [23] A. K. Jain, S. Prabhakar, and L. Hong, "A multichannel approach to fingerprint classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, pp. 348–359, 1999.
- [24] M. R. Girgis, A. A. Sewisy, and R. F. Mansour, "A robust method for partial deformed fingerprints verification using genetic algorithm," *Expert Systems with Applications*, vol. 36, no. 2, pp. 2008–2016, 2009.
- [25] A. Ross and A. Jain, "Biometric sensor interoperability: A case study in fingerprints," in *Biometric Authentication*. Springer, 2004, pp. 134–145.
- [26] C. Xinjian, T. Jie, Y. Xin, and Z. Yangyang, "An algorithm for distorted fingerprint matching based on local triangle feature set," *Information Forensics and Security, IEEE Transactions on*, vol. 1, no. 2, pp. 169–177, 2006.
- [27] S. P. Pissis. (2015) Acsmf implementation. [retrieved: 01.2016]. [Online]. Available: <https://github.com/solonas13/asmf>
- [28] C. Barton, C. S. Iliopoulos, and S. P. Pissis, "Fast algorithms for approximate circular string matching," *Algorithms for Molecular Biology*, vol. 9, no. 1, pp. 1–10, 2014.
- [29] NIST. (2015) Biometric special databases and software. [retrieved: 01.2016]. [Online]. Available: http://www.nist.gov/itl/iad/ig/special_databases.cfm
- [30] OpenCV-code. (2015) Implementation of guo-hall thinning algorithm. [retrieved: 01.2016]. [Online]. Available: <http://opencv-code.com/quick-tips/implementation-of-guo-hall-thinning-algorithm/>
- [31] A. Perez-Diaz and I. Arronte-Lopez, "Fingerprint matching and non-matching analysis for different tolerance rotation degrees in commercial matching algorithms," *Journal of applied research and technology*, vol. 8, no. 2, pp. 186–199, 2010.
- [32] C. Barton, C. S. Iliopoulos, S. P. Pissis, and F. Vayani, "Accurate and efficient methods to improve multiple circular sequence alignment," submitted.