

Development of Core-based Minutiae Triplets Model for an Improved Fingerprint Matching

Adegoke M.A.

Bells University of Technology
Department of Computer Science & Information
Technology
Ota, Nigeria

Adewumi O.A.

Bells University of Technology
Department of Computer Science & Information
Technology
Ota, Nigeria

ABSTRACT

Identifying distorted fingerprint images is a major problem in fingerprint recognition system. Distorted images can be caused by a change in scale of fingerprint images at capturing stage and at identification or verification stage. In this paper, distortion that may arise due to change in scale is addressed using an improved matching algorithm described as core-based minutiae triplets. This proposed algorithm uses minutiae triplets as a basis for fingerprint matching. For each minutiae point, a minutiae triplet is created which includes the core point and the closest minutiae to the core point. To make the proposed algorithm more effective, some optimizations were included so as to discard non-matching minutia triplets without comparing the whole representation. From experimentations carried out using the same data set, results showed that the core-based minutiae triplets algorithm had an average percentage accuracy of 98.25% while the minutiae inter-distance-based algorithm had an average percentage accuracy of 55.5%. Therefore, the proposed algorithm in the paper outperformed the minutiae inter-distance-based algorithm in terms of accuracy by 42.75%.

General Terms

Fingerprint Biometric Scanner

Keywords

Minutiae points, Minutiae inter-distance, Minutiae triplets, Core-based minutiae triplets, FMR, FNMR, Minutiae matching.

1. INTRODUCTION

Biometric recognition has been widely used for identity management of individuals in organizations and other public and private institutions with fingerprint emerging as the dominant one [1]. It has been reported that fingerprint-based biometrics is the most reliable and proven technique having superiority over iris, voice, face, gait and signature based biometric recognition systems [2]. Fingerprints are the marks made by ridges and valleys on the surface of a fingertip [2], [3] where ridges are the curved segment of marks and a valley is the region between two adjacent ridges. The flow of ridges on the fingerprint creates a pattern which is unique to an individual [4].

The global patterns include whorl, arch, tetrarch, right loop, left loop and double loop as shown in Figure 1.

Fingerprints also contain local ridge patterns, which are patterns that form on the ridges of the fingerprint. The local ridge patterns do not run evenly across our fingertip but display a number of characteristics called minutiae [5]

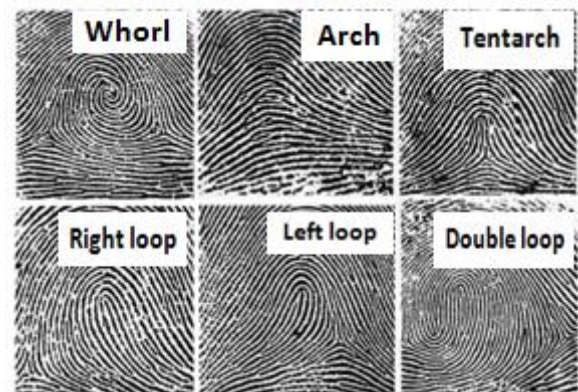


Figure 1: Global level features of a fingerprint

Minutiae of a fingerprint consists of the endpoints and crossing points of ridges; where an endpoint is the point where a ridge ends abruptly and cross point is the ridge point where a ridge bifurcates into two ridges. The minutiae ending and bifurcation are shown in Figure 2.

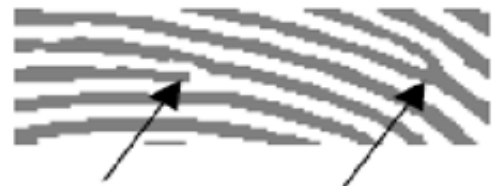


Figure 2: Minutiae Ending and Bifurcation

Fingerprint pattern matching is performed when the need arises to ascertain the source of a fingerprint image; this involves finding correspondences between a query fingerprint and a template fingerprint, and then generate a matching score [6]. The matching scores are expected to be high when the two fingerprints are from the same person and low otherwise.

Fingerprint matching faces a number of challenges including large intra-class variations and large inter-class similarity. Intra-class variations occur when there are variations in the fingerprint images of the same finger, while inter-class similarity occurs when there are similarities between fingerprint images from different fingers. Intra-class variations are caused by unequal pressure, orientation, magnification and translational placement with respect to the sensor [7]. It can also be caused by changes in the condition of the finger such as skin dryness, wetness or cuts. Inter-class similarity on the other hand can be as a result of limited

number of the global fingerprint patterns such as the arch, loop and whorl that are extracted from both the query and template fingerprints [5].

Fingerprints pattern matching methods include the image-based pattern matching, filter-based algorithms, minutiae-based and a combination of minutiae and image-based matching. In the image-based fingerprint [8] the global fingerprint images are transformed into a sequence of points in the angle-curvature domain. These patterns are compared for the query fingerprints and the template ones in the database. The matching between the query and the template is based on the least-square error of the Euclidean distance between corresponding points in the angle curvature domain. The filter-based algorithm [7] uses a bank of Gabor filters to capture both local and global details of a fingerprint as a compact fixed length code. However, it is much easier to extract minutiae patterns from low quality images than to extract ridge patterns.

The minutiae-based pattern matching compares only the minutiae of a query fingerprint with that of the template. If the minutiae of the two fingerprints are the same, a matching score is recorded. It has been proven that the minutiae pattern of each person is unique and does not change throughout one's life; and in case of injury or mutilation, they reappear within a short period [9]. Minutiae pattern is therefore one of the most important factors that can be used to determine whether two fingerprints are from the same person. In this work therefore, instead of bringing the entire ridge patterns into consideration, only the minutiae points that are extracted from both the template and query fingerprints are compared.

Some existing local structures for minutiae matching are the minutiae triplet [10], [11]. The minutiae triplet structure for minutiae matching and indexing is a geometric approach which consists of three minutiae points forming a triangle with three sides say $S1$, $S2$, and $S3$; where the three angles may be represented by $Q1$, $Q2$ and $Q3$. The minutiae triplet algorithm takes a fingerprint image of a query fingerprint, locate each of the m -triplets of the query fingerprint and compare with the template fingerprint images in the database. The m -triplet matching technique finds correspondence of m -triplets of the query fingerprint and the template fingerprints. Matching score is recorded when there is a match of the m -triplets of the query fingerprint in the database. It has however been reported that distortion arising from deformation in fingerprint images of the same fingerprint often affects matching accuracy [12], [17], [18]. It was further reported that m -triplet cannot handle distorted images [19]. This is because each of the distances $S1$, $S2$ and $S3$ of the m -triplet are counts of ridges between the three vertices. Deformation of ridges caused by skin dryness or wetness causes the lengths $S1$, $S2$ and $S3$ of the minutiae-triplet (m -triplets) to be distorted, making matching of a query fingerprint with that of the corresponding template difficult [19]. The distortion of the ridges also affects the local orientation of the m -triplets vertices. To overcome this problem and improve matching [18] uses Euclidean distance between every minutiae point and a core point of a fingertip in a fingerprint image to determine the minutiae matching scores of fingerprint images. The formulation of the algorithm was motivated by the fact that the relative distance of each minutiae to the core point does not change irrespective of the image directional flow. The algorithm was reported to have solved fingerprint matching problem that may arise due to image orientation. It was however reported that the algorithm is only suitable for a

specific image size. Once the scale of the fingerprint image differs from that of the template image, the accuracy of the matching algorithm is affected. The algorithm is therefore suitable or performs well when the same fingerprint scanner that was used for capturing is also used for verification or identification. This may not be practicable in this era of distributed systems when different scanners with different makers and magnification are available across the web. There is a need to develop a fingerprint matching algorithm that will be tolerant to deformation that may arise as a result of change in scale or magnification of the fingerprint images. Such a fingerprint matching algorithm will give room for interoperability by allowing fingerprint scanners that were different from the scanner that was used to capture the images to be used for verification and / or identification.

2. PROPOSED FINGERPRINT PATTERN MATCHING ALGORITHM

The algorithm proposed in this paper is motivated by the need to address the matching problems due to variation in the size (scale) of fingerprint images that may occur during image capturing and at verification or identification time. The new algorithm connects every two consecutive minutiae points to the core point to form a triangle as shown in Figure 3. The algorithm is premised on the fact that irrespective of the magnification of the scanner used for image capturing and verification, the angle formed by the two consecutive minutiae point with the core point is fixed [13]; so are the lengths of each of the minutiae points to the core point. The matching of two fingerprints is based on the size of the angle that two consecutive minutiae points formed with the core point which is fixed irrespective of the magnification or specification of the fingerprint scanner. The fact that the Euclidean and spatial relationships between the core point and minutiae points of fingerprint images are fixed irrespective of the orientation [18], the proposed algorithm uses the power of congruency and the relationships that exists between the Euclidean and spatial distances of a core point and minutiae points to improve intra-class matching error that may arise due to fingerprint scanner magnification. The new algorithm is described as core-based m -triplet algorithm. The triplets consist of every two consecutive minutiae points and the core point as shown in Figure 3. This is based on the Side Angle Side (SAS) theorem [14].

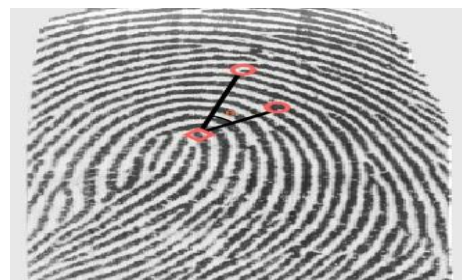


Figure 3: Fingerprint image showing the angle formed in the minutiae triplet

3. CORE-BASED MINUTIAE TRIPLETS MATCHING ALGORITHM

The proposed algorithm is based on minutiae-triplets representation. It matches an input image to a template image using correspondences of the angle formed at the core point in the minutiae triplets.

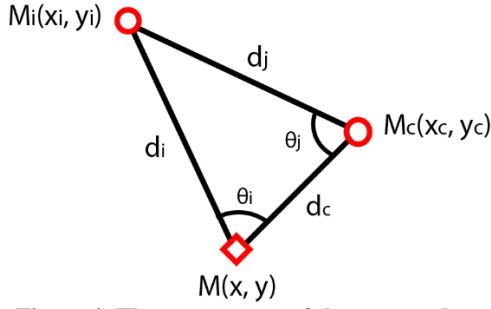


Figure 4: The components of the proposed matching algorithm

Given a fingerprint image described by the minutiae set M we compute the m -triplets as follows: For each $m \in M$, we build its m -triplet that includes \mathbf{p} , the Euclidean distances between the three minutiae points and the angles formed by the minutiae triplet with the core point. This way of computing m -triplets makes the proposed algorithm tolerant to the magnification that may arise, due to the type and size of scanner used across the web.

The proposed algorithm consists of three major steps:

Step 1: Let Q and P be the query and template fingerprint minutiae set respectively. Create query and template m -triplets sets R and T respectively. For each $\mathbf{r}_i \in R$, let \mathbf{r}_i contain $\{\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, \theta_1, \theta_2, \theta_3\}$. Where $\mathbf{d}_1, \mathbf{d}_2$, and \mathbf{d}_3 represent the Euclidean distances between the three minutiae points and $\theta_1, \theta_2, \theta_3$ represent the angles formed by the minutiae triplet.

Step 2: Let $A \leftarrow \{\}$ be the set that will contain local matching m -triplets pairs. For each query m -triplet $\mathbf{r}_i \in R$, perform linear search looking for the template m -triplets $\{\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_n\} \subset T$ with Euclidean distance and angular distance not exceeding threshold, discarding duplicates and add the pairs $(\mathbf{r}_i, \mathbf{t}_n), (\mathbf{r}_i, \mathbf{t}_n), \dots, (\mathbf{r}_i, \mathbf{t}_n)$ to A .

Step 3: The similarity score is calculated with the formula:

$$S = \frac{n^2}{|Q|*|P|}, \text{ where } n \text{ is the number of matching minutiae pairs, } Q \text{ and } P \text{ are the query and template fingerprint minutiae sets respectively and } S \text{ is the similarity score obtained from the match.}$$

The fingerprint images are considered a match if the similarity score $S \geq t_s$. Threshold t_s is a parameter of the algorithm, and must be tuned based on the image characteristics.

The core-based minutiae triplets' algorithm is given below:

Input: Inputfinger (minutiae table for input finger), Templatefinger (minutiae table for template finger) and database (the fingerprint database the algorithm uses).

Output: Similarity score between the input finger and the template finger.

Process: Get_similarity_score (database, inputfinger, templatefinger, margin1, margin2)

```
m = load(database);
inputfinger = m.(inputfinger);
templatefinger = m.(templatefinger);
scorearray = NULL;
for i=1:size(inputfinger,1)
```

```
temp_check = NULL;
for j=1:size(templatefinger,1)
    check = NULL;
    if (isempty(scorearray)==0)
        check = find(scorearray(:, 2)==
                    templatefinger(j,1));
    end
    if (check==NULL)
        abs_d1 = |inputfinger(i,2)-templatefinger(j,2) |;
        abs_d2 = |inputfinger(i,3)-templatefinger(j,3) |;
        abs_d3 = |inputfinger(i,4)-templatefinger(j,4) |;
        abs_t1 = |inputfinger(i,5)-templatefinger(j,5) |;
        abs_t2 = |inputfinger(i,6)-templatefinger(j,6) |;
        abs_t3 = |inputfinger(i,7)-templatefinger(j,7) |;
        if (abs_d1 <= margin1 && abs_d2 <= margin1
            && abs_d3 <= margin1...
            && abs_t1 <= margin2 && abs_t2 <=
            margin2 && abs_t3 <= margin2)
            sd = 1 - max(abs_d1, max(abs_d2,
            abs_d3))/margin1;
            sa = 1 - max(abs_t1, max(abs_t2,
            abs_t3))/margin2;
            spart = 1 - ((1-sd)*(1-sa));
            temp_check = [temp_check; [inputfinger(i,1),
            templatefinger(j,1), spart]];
        end
    end
    if (temp_check== NULL)
        temp_check = sortrows(temp_check, -3);
        scorearray = [scorearray; [temp_check(1,1),
        temp_check(1,2)]];
    next i;
next j;
n = size(scorearray,1);
n1 = size(inputfinger,1);
n2 = size(templatefinger,1);
similarityscore = (n^2)/(n1*n2);
exit
```

4. EXPERIMENT AND RESULTS

In this section, the performance of the fingerprint matching algorithm using the core-based minutiae triplets is evaluated.

The system requirement for the implementation consists of the software requirements and hardware requirements. The software uses Windows 10 as the operating platform and MATLAB as the programming language and computing environment. The hardware requirements consist of the following:

Memory: 8GB RAM.

Processor: 2.16GHz processor.

4.1 Data Set

The data used to evaluate the proposed fingerprint matching algorithm is based on a collection of fingerprint image impressions obtained from a sample fingerprint database [15].

The database consists of two parts; the template part and the query database. The fingerprint images in the query part are the replicas of the fingerprint images in the template part. Thus, data from the query part serves as input to the algorithm.

The sample fingerprint database contains 100 different fingerprints. For each fingerprint image in the template database, there are 8 images containing 8 distinct impressions and orientations of the fingerprint. There was no constraint such as minimum quality of the image on the users during the enrolment.

4.2 Experimental Setting

During experimentation, each query fingerprint image was matched against its own template fingerprint image in the database for a minimum of eight (8) times using scanners of different specifications and makers and at different orientation and impressions. This is to test for the efficiency of the core-based m-triplets algorithm when used with scanners of different sizes, makers and specifications. Four different scanners used in this experiment are Albima UNITY, Aratek FRO900, Credence ID Trident and SMUFS Biometric. Attempt was also made to match each query fingerprint image with template images that are not its own, using scanners of different sizes, types and specifications. This is to test for the level of inter-class accuracy measure of the algorithm. Table 1 shows the average results of matching ten (10) query

fingerprint images eight times each with its own template image using scanners of different sizes, types, makers and specifications. The ten (10) query fingerprint images were randomly selected. The table shows the average similarity score generated by matching a query fingerprint with its own template finger as well as the time taken to match the fingerprints. If the similarity score of each match is higher than the pre-determined threshold value, it is considered a match and accepted. If the similarity score is less than the predetermined threshold value, the input image is rejected and no match is recorded. It was observed and recorded that the similarity score obtained for each match is higher than the pre-determined threshold value. It should be noted that the predetermined threshold value is 0.5. This value is chosen because it is the standard value used in similar experiments [16]. The experiment was carried out with ten (10) groups of ten (10) query fingerprint images that were randomly selected from the 100 different fingerprint images. Each query fingerprint in each of the groups was matched with its template fingerprint eight (8) times. This was repeated for each of the ten (10) groups resulting in a total of 800 matching outcomes.

Table 1: Average Results for 10 random matching processes of a query fingerprint image using its corresponding image core-Based minutiae triplets' algorithm.

Name of Scanner	Input Fingerprint	Template Fingerprint	Average Similarity score	Status	Matching time (ms)
Abilma <u>UNITY</u>	Fingerprint 1_1	Fingerprint 1	0.69	Match	173
Aratek <u>FRO900</u>	Fingerprint 1_2	Fingerprint 1	0.72	Match	169
Credence ID <u>Trident</u>	Fingerprint 1_3	Fingerprint 1	0.62	Match	152
SMUFS Biometric	Fingerprint 1_4	Fingerprint 1	0.54	Match	140
Abilma <u>UNITY</u>	Fingerprint 1_5	Fingerprint 1	0.69	Match	140
Aratek <u>FRO900</u>	Fingerprint 1_6	Fingerprint 1	0.64	Match	145
Credence ID <u>Trident</u>	Fingerprint 1_7	Fingerprint 1	0.74	Match	174
SMUFS Biometric	Fingerprint 1_8	Fingerprint 1	0.69	Match	141

Table 2 shows the average results for randomly matching 10 query fingerprints eight (8) times each, with template fingerprint images that are not their own using the scanners of

different sizes, types and specifications. The similarity scores obtained for each match is below the predetermined threshold value so is the average similarity scores obtained. Thus, no match was recorded for this experiment as shown in Table 2

Table 2: Average Results for 8 random matching processes between different orientations of different fingerprints using core-based minutiae triplets algorithm

Sample	Input Fingerprint	Template Fingerprint	Average Similarity score	Status	Time Complexity (ms)
Abilma <u>UNITY</u>	Fingerprint 1_1	Fingerprint 2	0.40	Not a match	193
Aratek <u>FRO900</u>	Fingerprint 1_1	Fingerprint 3	0.49	Not a match	179
Credence ID <u>Trident</u>	Fingerprint 1_1	Fingerprint 4	0.41	Not a match	174
SMUFS Biometric	Fingerprint 1_1	Fingerprint 5	0.31	Not a match	211
Abilma <u>UNITY</u>	Fingerprint 1_1	Fingerprint 6	0.23	Not a match	181
Aratek <u>FRO900</u>	Fingerprint 1_1	Fingerprint 7	0.33	Not a match	173
Credence ID <u>Trident</u>	Fingerprint 1_1	Fingerprint 8	0.25	Not a match	153
SMUFS Biometric	Fingerprint 1_1	Fingerprint 9	0.33	Not a match	162

Table 3 compares the average accuracy of the core-based minutiae triplets' algorithm with the existing minutiae based algorithms that have used the same dataset. The core-based m-triplets algorithm presented in these papers outperforms the two related algorithms in terms of accuracy and the time taken by the algorithm to respond to validation task.

The core-based m-triplets algorithm recorded an accuracy of 98.25% using False Match Rate (FMR) and False Non-Match Rate (FNMR) measure. The average time taken to respond to matching task is 154ms.

Table 3: Average Percentage Accuracy of Algorithms.

Algorithm	Average Time (s)	Accuracy (%)
(Iwasokun et. al., 2013) algorithm using Minutiae Inter-distance	284	55.5
(Parziale et al., 2004), fingerprint matching technique using minutiae triangulation.	205	69.1
Proposed algorithm using Minutiae-triplet	154	98.25

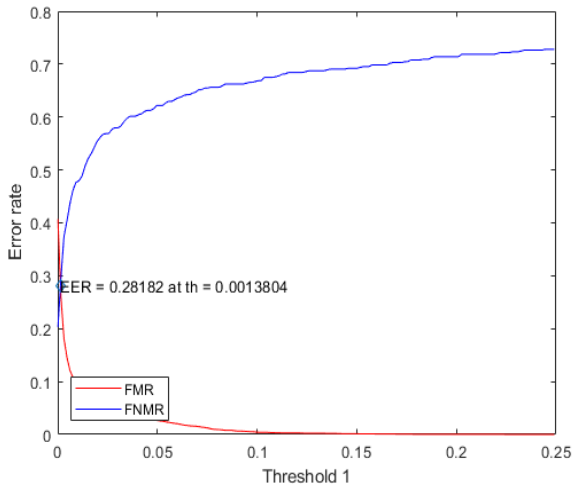


Figure 5: Graph showing FMR and FNMR curves for core-based minutiae triplets' algorithm.

Figure 5 is a graphical representation of comparing the FMR and FNMR of the core-based m-triplets algorithm with the existing algorithms on the same data set. The point at which the rate of false matching is made between a query fingerprint and a template fingerprint that are not from the same person is depicted by the blue curve in the graph. While the lower red curve represents the false non-match rates. The Extended Entity Relationship (EER) is obtained as the intersection of the two (2) curves.

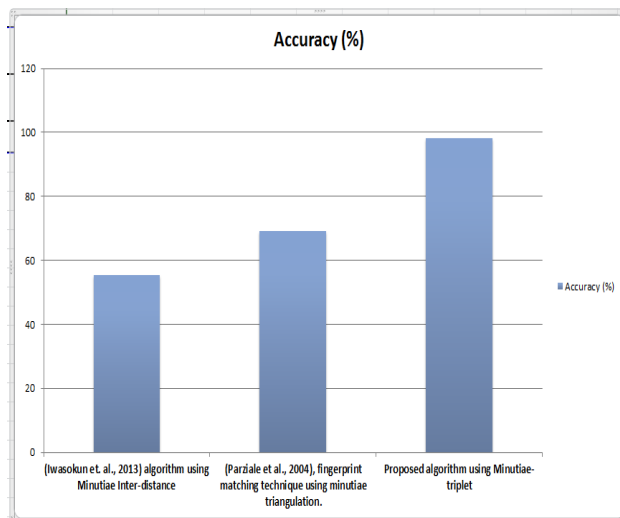


Figure 6: Graphical Comparison between the proposed core-based minutiae triplets' algorithm and other existing minutiae-based algorithms.

Figure 6 is a graphical representation of the comparison of the core-based minutiae triplets with other algorithms using bar chart. It is observed that the core-based minutiae triplets' algorithm has the highest accuracy of 98.5% as compared to the minutiae triangulation based algorithm which has an accuracy of 69.1%. However, the minutiae inter-distance based algorithm has the least accuracy of 55.5%.

4.3 Discussion of Results

It is observed from Table 3 that the proposed core-based minutiae triplets' algorithm out performs the minutiae inter-distance based algorithm proposed by [1] and the minutiae

triangulation based matching algorithm [20] in terms of accuracy and time taken for the matching to occur. This improvement can be justified by the use of the core point as a key parameter in the formation of the image formed by the query fingerprint. Using the power of congruency, the image of the query fingerprint will be congruent and the same with the image picked by scanners irrespective of the size and type of the scanner.

5. CONCLUSION

In this paper, a core-based minutiae triplet has been developed for the purpose of achieving invariance fingerprint matching between a query fingerprint and the template fingerprint of the same person. This algorithm hinged on the premise that regardless of the orientation and scale of the fingerprint image, the angles formed by the minutiae triplets would remain constant. The algorithm was tested using dataset from FVC 2002 fingerprint database. Results from experimentation showed that the proposed algorithm outperforms the existing algorithms that have used the same data. Future investigations can be carried out on filters whose main aim is to enhance fingerprint images in order to further improve the accuracy of the proposed algorithm.

6. REFERENCES

- [1] Iwasokun G.B, Akinyokun O.C and Angaye C. 2013. Article: Fingerprint Matching using Neighbourhood Distinctiveness. *International Journal of Computer Applications* 66(21):1-8, March 2013
- [2] Kumar, D. A. & Begum, T. U. S. 2013. A Comparative Study on Fingerprint Matching Algorithms for EVM. *Journal of Computer Sciences and Applications*, 1(4), 55-60.
- [3] Medina-Pérez, M. A., García-Borroto, M., Gutierrez-Rodríguez, A. E., & Altamirano-Robles, L. 2012. Improving fingerprint verification using minutiae triplets. *Sensors*, 12(3), 3418-3437.
- [4] Ganapati M.T. 2013. Identification of an individual through fingerprints. *J. Biol. Innov.* 2013;2(02):59–72
- [5] Zaeri N. 2011. Minutiae-based Fingerprint Extraction and Recognition, Biometrics, Jucheng Yang, IntechOpen, DOI: 10.5772/17527.
- [6] Iwasokun G.B 2015. Fingerprint Matching Using Minutiae-Singular Points Network. *International Journal of Signal Processing, Image Processing and Pattern Recognition* Vol. 8, No. 2 (2015), pp. 375-388
- [7] Jain A.K, Jianjiang F, Karthik N. (2013) FINGERPRINT MATCHING. Published by the IEEE Computer Society. 0018-9162/10/\$26.00 © 2010 IEEE
- [8] Saleh A., Ayman B. and Wahdan A. 2011. Fingerprint Recognition, *Advanced Biometric Technologies*, Girija Chetty and Jucheng Yang, IntechOpen, DOI: 10.5772/23476.
- [9] Iwasokun, G.B 2015. Fingerprint Matching Using Minutiae-Singular Points Network. *International Journal of Signal Processing, Image Processing and Pattern Recognition*. 8. 375-388. 10.14257/ijsp.2015.8.2.35.
- [10] Germain R.S., Andrea C., And Scott C. 1997. Fingerprint Matching Using Transformation Parameter Clustering. *IEEE COMPUTATIONAL SCIENCE & ENGINEERING*, IBM Thomas J. Watson Research Center
- [11] Bhanu B. & Tan X. 2003. Fingerprint indexing based on novel features of minutiae triplets. *Pattern Analysis and Machine Intelligence*, IEEE Transactions on. 25. 616-622. 10.1109/TPAMI.2003.1195995.

- [12] Jain, A.K, Ross, A., & Prabhakar, S. 2001. Fingerprint matching using minutiae and texture features. In Image Processing, 2001. Proceedings. 2001 International Conference on (Vol. 3, pp. 282-285). IEEE.
- [13] Roli B., Priti S. and Punam B. 2011. Minutiae Extraction from Fingerprint Images - a Review. *IJCSI International Journal of Computer Science Issues*, Vol. 8, Issue 5, No 3, September 2011 ISSN (Online): 1694-0814 www.IJCSI.org
- [14] Dokai M.C 2014. A Proof of Euclid's SAS (side angle side) Theorem of Congruence of Triangles via the Cross Section of a Double Cone. *IOSR Journal of Mathematics (IOSR-JM)* e-ISSN: 2278-5728, p-ISSN:2319-765X. Volume 10, Issue 1 Ver. III. (Feb. 2014), PP 26-33
- [15] FVC2002 (Fingerprint Verification Competition) <http://bias.csr.unibo.it/fvc2002>. © 2001-2002 BioLab - University of Bologna
- [16] Aju O.G, Orimoloye S.M, Omomule T.G (2018). An Enhanced Minitiae-based Fingerprint Matching Algorithm. *International Journal of Computer Applications (0975-8887)* Volume 182 – No. 34, December 2018.
- [17] Greenberg, Shlomo & Aladjem, Mayer & Kogan, Daniel & Dimitrov, Itshak. 2002. Fingerprint image enhancement using filtering techniques. Proceedings - International Conference on Pattern Recognition. 15. 322-325 vol.3. 10.1109/ICPR.2000.903550.
- [18] Iwasokun, G. B., Akinyokun, O. C., & Dehinbo, O. J. 2014. Minutiae inter-distance measure for fingerprint matching. In *International Conference on Advanced Computational Technologies and Creative Media (ICACTCM 2014)* (pp. 14-15).
- [19] Iloanusi O.N, 2011. Comparison of The Minutiae Quadruplets and Minutiae Triplets Techniques. *Nigerian Journal of Technology*, Vol. 30, No. 3, October 2011.
- [20] Parziale G, Niel A. (2004) A Fingerprint Matching Using Minutiae Triangulation. In: Zhang D., Jain A.K. (eds) *Biometric Authentication. ICBA 2004*. Lecture Notes in Computer Science, vol 3072. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-25948-0_34.