Minutiae Based Fingerprint Verification using Graph Model

Sonali Sen, Deyashini Bhattacharya, Soumili Dey, Sabarna Nandy

Abstract: Fingerprints offer one of the most reliable biometric traits that can be used for uniquely identifying a person. This proposed work demonstrates the use of graph theory in the field of fingerprint identification, in which a fingerprint is casted to a weighted complete graph and a weight matrix of this graph is used to describe the regions in the image and then checked for biometric authentication without considering Henry's classes. It further implements the concept of graph isomorphism along with edge mapping for matching of fingerprints which portrays the potential of graph-based methods for fingerprint representation, storage, and matching. The proposed algorithm is robust to non-linear distortion, rotation and scaling. The algorithm is tested on a database of Fingerprint Verification Competition (FVC) and has been found to be an efficient and a reliable one as compared to image processing which deals with the entire image for comparison between two fingerprints using pattern recognition.

Keywords : Minutiae, Graph Isomorphism, Sub Graph Isomorphism, Integer Generalized Bresenham Line Draw Algorithm, Fingerprint.

I. INTRODUCTION

Fingerprint being an immutable and easily available trait of biometrics, offers an infallible means of personal identification. Human fingerprints are rich in details called minutiae; extraction and proper mapping of which serve as the basis of biometric identification of an individual. Fingerprint recognition includes two sub-domains: one is fingerprint verification (One-to-one matching) and the other is fingerprint identification (One-to-many matching) as shown in Fig.1. Fingerprint identification problems following the pattern recognition techniques, requires combination of several processes in order to increase the accuracy and reliability of the system. On the other hand, the approach of representation and authentication of fingerprint discussed in this work makes use of graphs in which a simpler and reliable solution to the problem of representation and storage of a fingerprint with the minutiae details has been suggested.

The central idea of this work is to suggest an algorithm for fingerprint authentication, which, could serve as an improvement over the existing pattern-based matching techniques. In order to achieve that, the use of weighted undirected complete graph has been made to represent the fingerprints. The graphs are represented as edge weight matrices depending upon the application area. As a bi-product, the space required for the storage of the fingerprints in the database have also been substantially reduced as a matrix of reduced size of the order of the number feature points, is stored for future reference rather than a complete image.

An algorithm for counting the number of intersecting ridge lines between two minutiae points have also been proposed, which can be used for weight representation. Furthermore, the complexity of fingerprint matching also reduces due to implementation of graph isomorphism check on the graphs rather than processing the complete image. The concept of sub-graph isomorphism with edge-weight correspondence is able to detect a partial match between the fingerprints. The algorithm also implements the concept of threshold for matching fingerprints, which, is determined using reliable statistical values. This is useful in the forensic situations where fragment fingerprint may contain noise. The algorithm so proposed is independent of distortion, rotation and transformation along with secure storage and is computationally cheaper without affecting the authenticity and reliability. Distortion changes both geometric position and orientation, and leads to difficulties in establishing a match among different impressions acquired from the same fingertip. This drawback has been overcome in the graph theoretical approach through sub-graph isomorphism so that even if the fingerprint acquired for verification is distorted and is not a perfect match with the one present in the database, the algorithm can detect a match if they have been acquired from the same fingertip. Thus, in such real life applications where the print acquired is not of a high quality, this algorithm is suitable to be used.

The Proposed work is described in Section 2, the Algorithm Design in Section 3, Section 4 is used to describe the Results and Section 4 is for Conclusion.

Revised Manuscript Received on October 30, 2019.
* Correspondence Author
Sonali Sen*, Department of Computer Science, St. Xavier's College, Kolkata, India. Email: sonalisen@sxccal.edu
Deyashini Bhattacharya, Department of Computer Science, St. Xavier's College, Kolkata, India. Email: bhattacharyageorge@gmail.com
Soumili Dey, Department of Computer Science, , St. Xavier's College, Kolkata, India. Email: soumilidey98@gmail.com
Sabarna Nandy, Department of Computer Science, St. Xavier's College, Kolkata, India. Email: nandy.sabarna@gmail.com

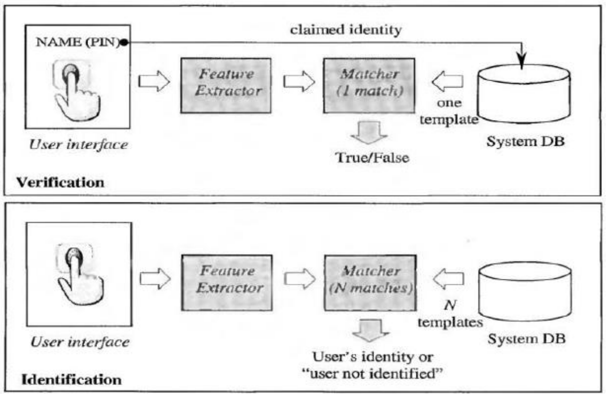


Fig. 1: Identification Vs Verification [1]

II. PROPOSED WORK

A fingerprint recognition system constitutes of Fingerprint acquiring device for generating digital image of fingerprint, Minutia Extractor and Minutia Matcher as shown in the Fig.2 below. The work suggests an alternative for the partial identification where the use of graphs has been made to store the information about the various traits of a fingerprint along with their geometric neighborhood as a weight matrix which is shown in Fig.3.

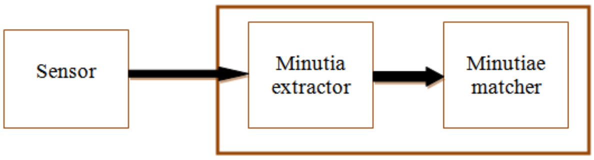


Fig.2: Basic structure of Fingerprint Recognition System [1]

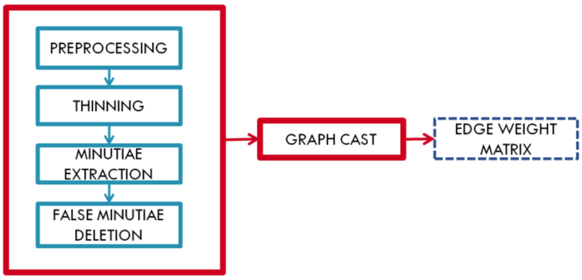


Fig.3(a) : First phase of the algorithm

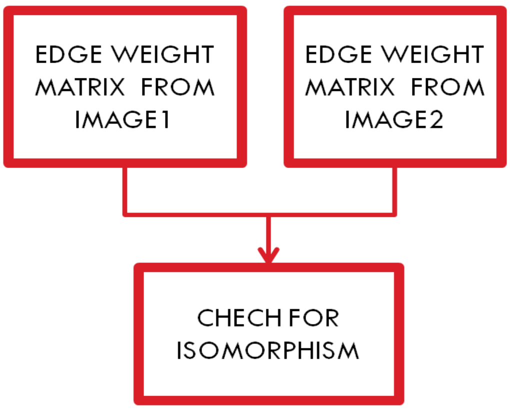


Fig 3(b) : Second phase of the algorithm

A. Pre-processing

The various pre-processing steps include image enhancement, binarization, segmentation, ridge thinning, and noise reduction.

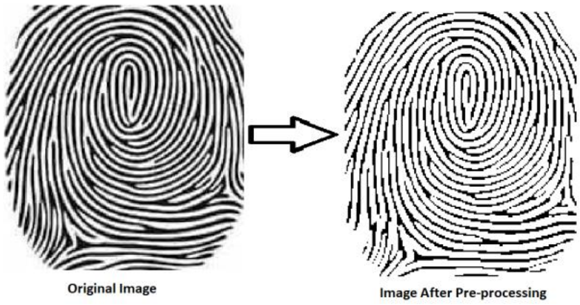


Fig. 4 :Original Image Vs Pre-processed Image

B. Minutiae Extraction

After the fingerprint pre processing, identifying and marking the minutia points is the next most important step. The red dots show termination points, and the blue dots show bifurcation points.

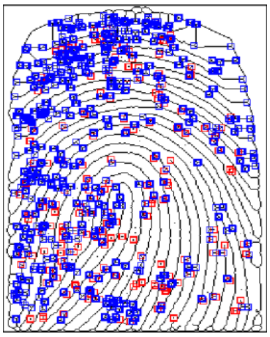


Fig. 5 : Image with minutiae points marked

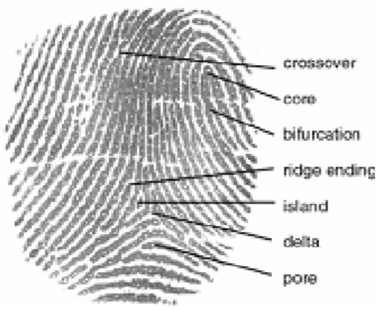
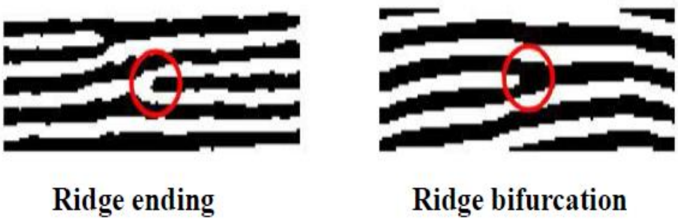


Fig. 6: Ridge ending & Bifurcation Pattern used for minutiae extraction



C. False Minutiae Identification and Removal

The pre-processing stage does not usually fix the fingerprint image in total. For example, false ridge breaks due to insufficient amount of ink and ridge cross-connections due to over inking are not totally eliminated. These false minutiae will significantly affect the accuracy of matching if they are simply regarded as genuine minutiae. So removing false minutiae points was essential to keep the fingerprint verification system effective and efficient. A fingerprint image after deletion of the false minutia points have been shown in Fig. 7 below. The red dots show termination points, and the blue dots show bifurcation points.[3]

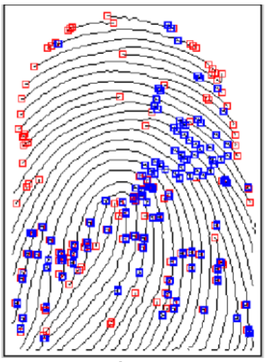


Fig. 7: Minutia points after removal of false minutiae

D. Construction Of Graph From The Extracted Minutiae Points

Determination of Edge Weight

In the proposed algorithm, the next step in the process is to cast the extracted minutiae points into a connected graph. A completely connected graph is constructed where each minutia is considered as a node and each node (minutia) is connected to every other node. The graph is represented in the form of an edge weight matrix. Thus, for *m*-number of minutiae points present in a fingerprint, a *m*-dimensional $(m \times m)$ weight matrix will be formed. The edges of the graph can be weighted in one of the three following ways:

$e(i,j)=d$=shortest distance between node i and node j determined using the coordinate geometry shortest distance formula $d=\sqrt{((x_2-x_1)^2 + (y_2-y_1)^2)}$ where (x_1,y_1) is the position of node i and (x_2,y_2) is the position of node j. This weight of graph is applicable if the input graph to be matched may or may not be of same dimension, may have undergone a change in orientation. Hence, we can perform an optimized matching without fixing the orientation of the input image as the graphs are rotation invariant.

Fig. 8 : edge weight (*e(i,j)*)

(i) $e(i,j)$=Number of ridge lines intersecting the straight line joining node i and node j. This method of allocating weight to edge is useful if the input fingerprint to be matched is scaled or magnified. In that case, the relative distance between the minutia points may change due to change of scale of the image, and thus it is no longer a reliable measure.

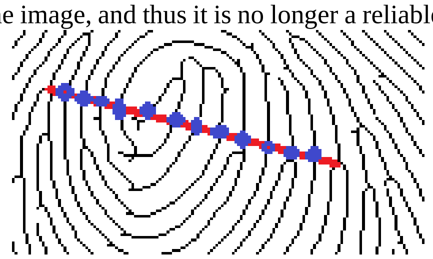


Fig. 9 : ridge lines intersecting the two bifurcations=12 (excluding the starting and ending pixels)

(i) $e(i,j)=m=\tan\theta$=slope of the straight line joining two minutiae points where the coordinates of the two points are received from the pixel position of minutiae points. However in case of rotation of the fingerprint, proper transformation needs to applied to angle depending on the degree of rotation about the origin. Hence the origin and degree of rotation must be determined.

Let (x_1,y_1) and (x_2,y_2) be the pixel position of two minutiae points

Hence edge weight = $m=\tan\theta=(y_2-y_1)/(x_2-x_1)$

On applying rotation of ϕ, the following transformation needs to be applied:

$y' = y\cos\phi - x\sin\phi$

$x' = y\sin\phi + x\cos\phi$

where (x',y') are the new coordinates after rotation and ϕ is the degree of rotation of fingerprint.

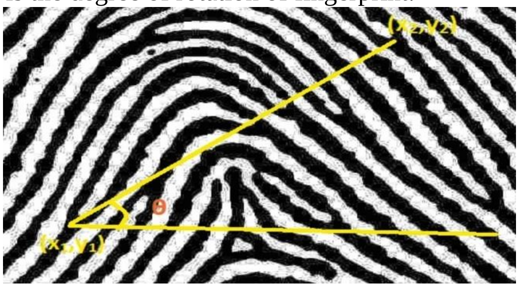


Fig. 10: Edge weight by detecting the slope of the straight line joining two points

The graph thus formed has the following properties:

- The weight matrix is symmetric as the edge considered is direction independent and symmetric, which means that the $e_{ij}=e_{ji}$.
- The graph is complete; as every node (minutiae point in the graph) is connected to every other node and their corresponding edge weight are calculated.
- The graph is weighted as each edge is assigned a weight depending on neighborhood characteristics like distance, slope or number of intersecting ridge lines.
- The graph is undirected as the edges are bidirectional.

Construction Of Graph

The minutiae point pixels containing ridge endings and bifurcations are represented by 1 and 2 respectively. These corresponding pixel positions identified as 1 or 2 are numbered row wise to represent the nodes. This is followed by construction of graph where the weight between each pair of nodes is calculated and represented as a complete graph in the form of a weight matrix.

Retrieval Number F8165088619/2019©BEIESP
DOI: 10.35940/ijeat.F8165.088619
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

The graph thus constructed is of the order of the number of minutiae points considered. The graph is in the form of a weight matrix and stored in the database. Since, the storage of a matrix requires lesser space than the storage of a picture, an optimization regarding requirement of storage space is also achieved through the use of this algorithm. This completes the enrolment process of the fingerprint in the database in the database after casting it into a graph data structure.

Let the minutiae extracted matrix be:

0	1	0	0
2	0	0	0
0	0	2	1
2	0	0	0
0	2	2	0
1	0	0	0
0	1	0	1

Then the matrix is numbered as:

0	1	0	0
2	0	0	0
0	0	3	4
5	0	0	0
0	6	7	0
8	0	0	0
0	9	0	10

Fig. 11: Illustration of extraction of minutiae

Let the edge weight be defined as the shortest distance between the two minutia point nodes connected by that edge. Let row 1 and column 1 represent the origin of the Cartesian coordinate system used for distance calculation.

Then the corresponding edge weight matrix created using $\sqrt{((x2-x1)^2+(y2-y1)^2)}$:

	1	2	3	4	5	6	7	8	9	10
1	0	1	2	3	3	4	4	5	6	6
2		0	2	3	2	3	4	4	5	6
3			0	1	2	2	2	4	4	4
4				0	3	3	2	4	4	4
5					0	1	2	2	3	4
6						0	1	1	2	3
7							0	2	2	2
8								0	1	3
9									0	2
10										0

Fig. 12: Edge weight matrix of GC (candidate fingerprint)

Observation: The edge weight matrix formed is symmetric

Let the fragment minutiae points be from row 2 to 6 and columns 1 to 3

2	0	0
0	0	2
2	0	0
0	2	2
1	0	0

Then the corresponding edge weight matrix created is:

0	2	2	3	4	4
	0	2	2	2	4
		0	1	2	2
			0	1	1
				0	2
					0

Fig. 13: Edge weight matrix of GF(fragment fingerprint)

E. Fingerprint Matching Using Sub Graph Isomorphism

In the fingerprint matching problem, the concept of Graph Isomorphism is used as the backbone of the matching operation. However, the concept needs to be extended in such a manner such that the algorithm is able to detect a partial match between the questioned fingerprint and the candidate fingerprint based on some threshold value. This is necessary because the fingerprint images captured in real-time systems are always not identical, even if they come from the same fingertip. They are often distorted or geometrically varied due to variable pressure on the scanner, presence of dirt, oil and sweat on the fingertip, insufficient ink while capturing the fingerprint in hardcopy format, and due to many other such reasons. If exact isomorphic graphs are searched for and matched, then more often than not, genuine prints from the same fingertip will be rejected as a mismatch.

The problem of partial matching of fingerprints is solved through a concept called sub-graph isomorphism with edge weight correspondence, where a match operation is performed based on the relative position of the adjacent nodes of the graph and a match is declared based on some threshold value decided by rigorous experiment on a large data set. The original fingerprint is called the Candidate Fingerprint *GC (VC,EC)* and the fingerprint to be validated is called the Fragment Fingerprint *GF (VF,EF)* where isomorphism is implemented assuming $|VC|>=|VF|$. The isomorphism is checked using two proposed algorithm viz. Generalized Combinatorial Sub Graph Isomorphism and Sequential Combinatorial Sub Graph Isomorphism. The graph with a greater cardinality is chosen as the Candidate fingerprint. Isomorphism of *GF(VF,EF)* and *GC (VC,EC)* is determined hence determining a match or mismatch:

Match: If the input fingerprint is an exact copy of the candidate fingerprint then the weight matrix will definitely be the same or there will exist a mapping between original and transformed weight matrix, and so it will be isomorphic to the candidate fingerprint when their corresponding edges are mapped.

Mismatch: If the other fingerprint is not the same fingerprint as that of the candidate fingerprint then the weight matrix of the other fingerprint will never be same as that of the candidate fingerprint. A particular value of offset is chosen depending on the minimum number of minutiae and their corresponding neighborhood that must match for two identical fingerprints, and if the number of mismatch exceeds that offset, the two fingerprints can be concluded to be mismatched thus reducing the complexity from comparing all the minutiae points or entire pattern.

Here, $E=f$ (neighbourhood of minutiae).

Let transformation T is calculated such that $T(X) \subseteq S$, where X=fragment input fingerprint, S=candidate fingerprint.

Aim is to determine *GF* is isomorphic to a sub graph of *GC* or $X \subseteq S$, and to verify that, the minutiae which compare the two graphs represent the same fingerprint. If the T is equivalent for all mapping between EF and EC then the spanning tree of GF and GC are same and the graphs are isomorphic and the two fingerprints are declared to be a biometric match.

The result of match between two fingerprints depends upon:

(i) If $GF(VF,EF) \subseteq GC(VC,EC) \dashrightarrow \exists T \mid T(X) \subseteq S$

(ii) *GF* is isomorphic to a sub graph of *GC*

Retrieval Number F8165088619/2019©BEIESP
DOI: 10.35940/ijeat.F8165.088619
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Isomorphism is implemented using two proposed algorithm viz. Generalized Combinatorial Sub Graph Isomorphism and Sequential Combinatorial Sub Graph Isomorphism.

Generalized Combinatorial Sub Graph Isomorphism

Let EF be of dimension *rxr* and EC be of dimension *nxn* .In this algorithm all possible *rxr* sub graph combinations are extracted from the Candidate fingerprint *GC*, such that choosing of row *i* is accompanied by choosing of column i from the weight matrix, where $1 \leq i \leq m$ (m=order of weight matrix). Hence the total number of sub graphs extracted from GC is $^{n}C_{r}$ and all these possible sub graphs are checked for isomorphism with graph *GF* which determines the result. If any of the sub graphs of *GC* is isomorphic to *GF*, then the fingerprints show a match. Though this algorithm is more generalized and accurate, it has a comparatively higher complexity as compared to the algorithm in Sec. 2.5.2.

Sequential Combinatorial Sub Graph Isomorphism

In this algorithm instead of considering all possible *rxr* sub graph combinations, only the sequential combinations are considered. In graph representation of fingerprints, since the edge weights are dependent on topology and are functions of neighborhood characteristics, hence instead of taking all possible combinations of sub graphs, only sequential combinations are considered where the edge weights remain consistent and continuous in a particular *rxr* neighborhood. Since the weights remain consistent in an *rxr* matrix, hence only consecutive and sequential combinations are considered. Hence from the *nxn* Candidate weight matrix, only *n-r+1* number of sub graphs are taken for checking isomorphism with *rxr* Fragment weight matrix. Here for construction of sub graph, choice of row *i* is accompanied by choice of column *i* from weight matrix such that $1 \leq i \leq m$, followed by choosing row and column *j* where $j=i+1$, $2 \leq i \leq m-1$, and so on.

III. ALGORITHM DESIGN

A. Use Of Geometric Shortest Distance Between Two Nodes To Weight The Edges Of G (V, E)

Step 1 : Input minutiae extracted matrix of order *r* x *c*.
Step 2 : Number all minutiae points from 1 to *n* where minutiae locations are marked by 1 (for ridge ending) or 2 (for bifurcation).
Step 3 : Find row and column *(x1,y1)* of node *i* and row and column *(x2,y2)* of node *j*.
Step 4 : Find $d \leftarrow \sqrt{((x2-x1)^2+(y2-y1)^2)}$.
Step 5 : Put *d* in *i* th row and *j* th column of edge weight matrix where $e(i,j)=d$.
Hence the weight matrix of order *n* x *n* is obtained.

B. To find the number of ridge lines intersecting the shortest distance between node i and node j

Step 1 : Input minutiae extracted matrix of order *r* x *c*.
Step 2 : Number all minutiae points from *1* to *n*.
Step 3 : Find row and column *(x1,y1)* of node *i* and row and column *(x2,y2)* of node *j*
Step 4 : Find all the pixels on the shortest line joining node *i* and *j*, using Integer Generalised Bresenham Line Draw algorithm.
Step 5 : It is checked whether the extracted pixel is a 0 valued pixel or not. If so it is counted as an intersecting ridge. The 0 valued pixels are considered to be intersecting ridge lines between node *i* and node *j*. If all the pixels between node *i* and node *j* are scanned then go to step 6 else go to step 4.
Step 6 : The number of intersecting ridge lines between node *i* and node *j* and node *j* and node *i* may not be the same due to Integer Generalised Bresenham line draw algorithm. Hence the maximum of the ridge line count from node*(i,j)* and node*(j,i)* is considered. We store the maximum ridge line count in the edge weight matrix. If *i* and *j* are same then we store 0 in the edge weight matrix.
Hence the weight matrix of order *n* x *n* is obtained.

C. Sequential Combinatorial Sub-graph Isomorphism

To determine whether graph *GF(nxn)* and graph *GC(rxr)* are isomorphic:
Step 1 : Take a *Window_counter* and initialize it to 0.
Step 2 : Take number of consecutive sequences $^{n}C_{r} = n-r+1$.
Step 3 : Construct a 2-d array *brr* containing only the consecutive sequences among all possible combinations of $^{n}C_{r}$. Now, *brr (n-r+1,r)* <--- generate all consecutive sequences.
Step 4 : Chose *drr(i,j)* of size *rxr* which is a sub graph of *EC* such that $i,j \in brr(k,:)$. Repeat Steps 5,6,7 for all possible *k*, for all *n-r+1* sub graphs of *GC*.
Step 5 : Initialize *ctr_block* to 0.
Step 6 : Chose edge $a_{ij} : i,j \in drr$ and chose edge $b_{ij} : i,j \in EF$. Chose $T \mid T(a) \subseteq b$ for all $a \in EF$. Now if *T(a)* is isomorphic to *b* and the sub graph of *GC* is isomorphic to *GF* then increment *ctr_block*.
Step 7 : Check if *ctr_block >=r*r-offset*, where *offset* is determined by statistical methods. If yes, then increment *Window_counter* else result is *fail*.
Step 8 : If *fail > threshold* , conclude a mismatch. Here *threshold* is determined statistically.
Step 9 : If $Window\_counter >= \sqrt{(^{n}C_{r})}$ then GRAPHS ARE ISOMORPHIC else GRAPHS ARE NOT ISOMORPHIC.

D. Generalized Combinatorial Sub-graph Isomorphism

To determine whether graph *GF(nxn)* and graph *GC(rxr)* are isomorphic:
Step 1 : Take a *Window_counter* and initialize it to 0.
Step 2 : $^{n}C_{r}$ <--- all possible combinations of sub-graphs $SG(r) \subseteq GC(n)$.
Step 3 : Construct a 2-d array *brr* containing all possible combinations of $^{n}C_{r}$. $brr(^{n}C_{r},r)$ <--- generate all possible combination of indexes of $^{n}C_{r}$ in recursive approach.
Step 4 : Chose *drr(i,j)* of size *rxr* which is a sub graph of *EC* such that $i,j \in brr(k,:)$. Repeat Steps 5,6,7 for all possible *k*, for all *n-r+1* sub graphs of *GC*.
Step 5 : Initialize *ctr_block* to 0.
Step 6 : Chose edge $a_{ij} : i,j \in drr$ and chose edge $b_{ij} : i,j \in EF$. Chose $T \mid T(a) \subseteq b$ for all $a \in EF$. Now if *T(a)* is isomorphic to *b* and the sub graph of *GC* is isomorphic to *GF* then increment *ctr_block*.

Step 7 : Check if *ctr_block* >=*r***r-offset*, where *offset* is determined by statistical methods. If yes, then increment *Window_counter* else result is *fail*.

Step 8 : If *fail* > *threshold* , conclude a mismatch. Here *threshold* is determined statistically.

Step 9 : If *Window_counter*>=$\sqrt{({}^{n}C_{r})}$ then GRAPHS ARE ISOMORPHIC else GRAPHS ARE NOT ISOMORPHIC.

E. Threshold Determination

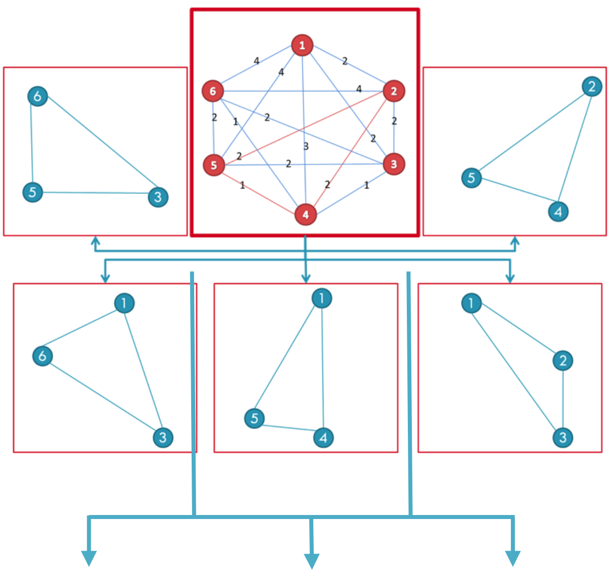
It is scientifically and biologically impossible for two persons to have 12 or more Galton detail in sequence. Anything less than this number, the analyst must be prepared to defend with supporting ridge features that are clearly visible. Locard also established if more than 12 concurring points are present, and the fingerprint is sharp, the certainty of identity is beyond debate. This level of reliability has been validated daily through the automated search upon trillions of fingerprints throughout the world. The offset for isomorphism process is determined on the basis of the above facts as follows:

- The threshold for number of matches in a (*r x r*) window as implemented in the sub graph matching is taken to be 13 as that is the minimum number of features that should match in any two fingerprints as mentioned above. Hence if 13 edge weights of candidate and fragment fingerprint are identical, the r x r block is considered to be matched.

- For larger number of minutiae points, which may include a larger number of undeleted false minutiae points, the offset, (that is the number of edges that must at least match between two fingerprints) is set as *2/3*r* , where r is the order of fragment fingerprint. This offset is determined experimentally considering a number of fingerprints.
  A threshold of the minimum number of *r x r* blocks that must match out of the n x n blocks is also required as all ${}^{n}C_{r}$ blocks do not match(especially for fingerprints obtained from crime scene). Hence offset is taken as square root of (${}^{n}C_{r}$).

IV. RESULTS

A. Illustration of all possible sub graphs from a given graph



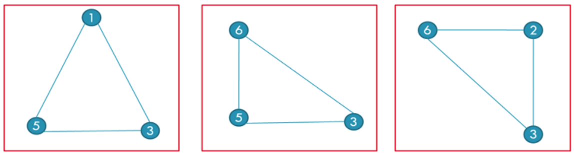


Fig 14:Illustration of some of the sub graphs of order *r* from a graph of order *n*. Here *n*=6 and *r*=3

4.2. Illustration of sub-graph isomorphism

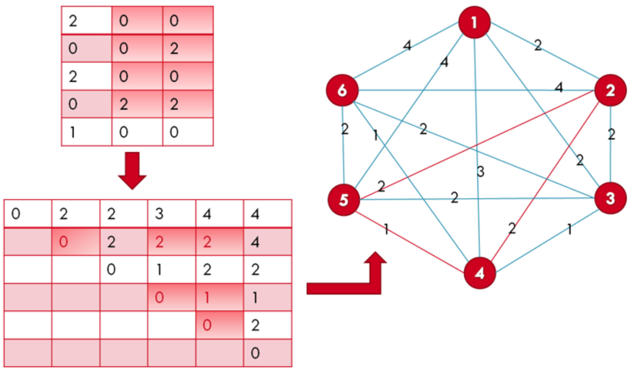


Fig. 15 : Construction of graph from candidate fingerprint GC(VC,EC)

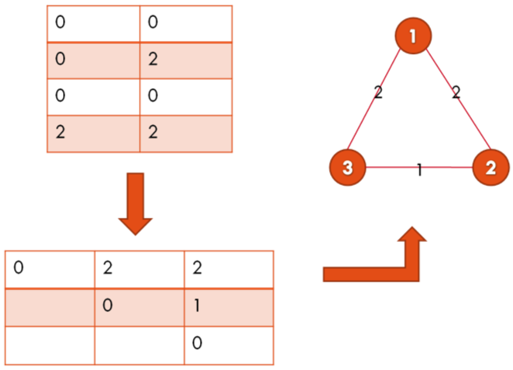


Fig.16 : Construction of graph from fragment fingerprint GF(VF,EF)

Now a particular fragment of the candidate is extracted to show that a particular sub graph of the candidate is isomorphic to the fragment fingerprint. Hence Fragment fingerprint is a sub graph of Candidate fingerprint and they are isomorphic

The edges of the graph *GF* is correspondent and equivalent to the edges of the graph *GC* which are marked as red, hence proving that *GF* is a sub graph of *GC* as $GF \subset GC$ and *GF* is isomorphic to a sub graph *GC* with mapped edge weights, where mapped weight of *GF, GC* ∊ *{2,2,1}*.

The proposed method helps in unique identification of a fingerprint

- The probability of finding two people with identical fingerprints is very small. In fact, no two identical fingerprints have ever been found same. Galton calculated that probability of finding identical prints was 1 in 64 millions. A second principle is that an individual's fingerprints do not change with time.

- The ridge events are commonly referred to as characteristics or minutiae, and their spatial relationship to one another in a friction ridge impression is the basis for fingerprint comparison and identification. Main proof of uniqueness is the spatial distribution of minutia points which is the basis of calculating edge in this algorithm.
- Since ridge flows are unique (no two areas of friction ridge skin are the same, not even on identical twins.) gradient gives a unique measure.
- It is scientifically and biologically impossible for two persons to have 12 or more Galton detail in sequence. That means if there are 12 or more minutia points matching then the two fingerprints are of the same finger.[Locard]
- Statistical studies show that Galton (level 2) ridge characteristics determined reliable thresholds to establish individuality. It has been reported that two different persons as a result of computerized search has ever been found to have anywhere near 12 Galton characteristics in agreement of level I and level II details in approximate relative position, characteristic type and ridge path direction (all of which is considered in the algorithm).Ultimately all algorithms for fingerprint matching guarantees precision of application but not the accuracy of conclusion, precision is determined by statistical methods and no scientific methods exist.

This proposed algorithm has been implemented on a large dataset of FVC and are transformed in various ways such as rotation, scaling, shifting ,cropping, for the purpose of investigating accuracy. The results of the fingerprint matching algorithm's performance are summarized in Table 1.

Table 1: Accuracy percentage of different transformed fingerprints by graph isomorphic algorithm.

Test Cases	Match Result (%)
Two exactly identical fingerprints	100
Scaled fingerprint	98
Rotated fingerprint	97.4
Cropped or Fragment fingerprint	98.5
Translated fingerprint	99.2

V. CONCLUSION

The algorithm suggested in the work is good enough to solve the fingerprint matching problem fairly accurately. The benefits of this algorithm are that it is rotation independent, allows secured transfer of a segment over the internet and easy to store in a database, allows comparison of weight matrix of two fingerprints of a smaller size as compared to an entire fingerprint image and no use of additional tools for image processing and feature detection. The algorithm can also process and detect match if the input is a partial copy of the candidate. Furthermore, since the use of graph data structure has been used to represent the fingerprint, although the initial pre-processing is a bit costly and complex, but once stored it takes far lesser space, which is a significant gain from the storage point of view because real-life fingerprint databases are usually enormous in size. Even on poor quality of fingerprints, where most of the pattern based algorithms fail, this algorithm can return some result based on the structure of the graph obtained, and thus it might be of practical use for the fingerprints obtained from crime scenes where the quality of prints are usually noisy and distorted. Processing of the fingerprints while matching it with an input fingerprint also becomes computationally less costly since the graph isomorphism operation used here is computationally simpler than pattern or image based matching operations. However this algorithm has only been implemented in existing database and as a future scope it can be extended to be implemented in a real time system.

REFERENCES

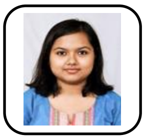
1. "Fingerprint Recognition for Person Identification and Verification Based on Minutiae Matching": Mouad.M.H.Ali ,Babasaheb Ambedkar, Babasaheb Ambedkar, Pravin Yannawar, Babasaheb Ambedkar, DOI: 10.1109/IACC.2016.69 , 2016 IEEE 6th International Conference on Advanced Computing, 18 August 2016, pp. 332-335.
2. "Fingerprint Verification System using Minutiae Extraction Technique": Manvjeet Kaur, Mukhwinder Singh, Akshay Girdhar, and Parvinder S. Sandhu , World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering Vol:2, No:10, 2008, pp. 3405-3408.
3. [3] "Removal of False Minutiae Using Fuzzy Rules" : Mr. N.Surya, Mr. K.BhavaNarayana, Mr. K.Sathish, Research Inventy: International Journal of Engineering And ScienceVol.3, Issue 6, August-2013, pp. 56-60.
4. "Fingerprint Recognition Using Minutia Score Matching" : Ravi. J , K. B. Raja , Venugopal. K. R., International Journal of Engineering Science and Technology Vol.1(2), 2009, pp. 35-39.
5. "A Comparative Analysis on Various Techniques of Fingerprint Recognition" : Manisha Sharma1, Deepa Verma, International Journal of Engineering Research & Technology (IJERT)Vol. 2, Issue 11, November - 2013, pp. 2099-2100.
6. " Ridge Enhancement in Fingerprint Image Using Oriented Diffusion": Robert Hastings, IEEE Computer Society On Digital Image computing Techniques and Applications, 07 January 2008 ,pp. 245-246.
7. "Fingerprint Singular Point Detection Based on Modified Poincare Index Method" : Gabriel Babatunde Iwasokun1,2 and Oluwole Charles Akinyokun3, ISSN: 2005-4254 IJSIP, International Journal of Signal Processing, Image Processing and Pattern Recognition Vol.7, No.5, October 2014 , pp. 259-272.
8. [8] "Graph-based and structural methods for fingerprint classification" : Gian Luca Marcialis, Fabio Roli, Alessandra Serrau, DOI:10.1007/978-3-540-68020-8_8, Applied Graph Theory in Computer Vision and Pattern Recognition, 1.1.2007, pp. 3-15.
9. "Human Identification and Fingerprints: A Review" : Nitin Kaushal and Purnima Kaushal, DOI: 10.4172/2155-6180.1000123, Journal of Biometrics & Biostatistics,January 2011, pp. 1-4.
10. "Fingerprint image enhancement and minutiae extraction algorithm": Letian Cao and Yazhou Wang, Master Thesis, 31-05-2016, pp. 1-23.
11. "Techniques of Binarization, Thinning and Feature Extraction Applied to a Fingerprint System" : Romulo Ferrer L. Carneiro , Jessyca Almeida Bessa , Jermana Lopes de Moraes , Edson Cavalcanti Neto , Auzuir Ripardo de Alexandria, Segmentation International Journal of Computer Applications (0975 8887)Volume 103 - No. 10, October 2014, pp.2-6.
12. "Fingerprints thinning algorithm" **:** V. Espinosa-Duro**,** Published in: IEEE Aerospace and Electronic Systems Magazine**,** DOI: 10.1109/MAES.2003.1232157, 15 September 2003**,** pp.28-30.
13. "Minutiae Extraction from Fingerprint Image -a Review": Roli Bansal ,Priti Sehgal, Punam Bedi, IJCSI International Journal of Computer Science Issues , Vol. 8, Issue 5, No 3, September 2011, ISSN (Online): 1694-0814, pp.79-81.
14. "Fingerprint recognition using MATLAB": Zain S. Barham, Dr. Allam Mousa, academia.edu, Graduation Project, 17-5-2011, pp. 14-17,22-45.
15. "Adaptive Crossing Numbers And Their Application To Binary Downsampling" : Decencière, Etienne & Bilodeau, Michel, Image Analysis and Stereology, 26.10.5566/ias.v26, ISSN 1580-3139, May,21,2007, pp. 73-81.

16. "Latent Fingerprint Matching: A Survey" **,** Anush Sankaran , Mayank Vatsa , Richa Singh , Publisher: IEEE , Electronic ISSN: 2169-3536 **,** INSPEC Accession Number: 14620830 DOI: 10.1109/ACCESS.2014.2349879, 20 August 2014, pp. 37-39**.**
17. "Graph Isomorphism applied to Fingerprint Matching " : Regina de Cássia Nandi, André Luiz Pires Guedes, Workshop On Graph Theory And Applications,November,20,2006, pp.1-3.
18. "Graph Theory with Algorithms and its Applications" : S. Saha Ray, DOI: 10.1007/978-81-322-0750-4_2, Springer India, 2013, pp. 5-15.
19. "A Comparative Analysis on Various Techniques of Fingerprint Recognition": Manisha Sharma, Deepa Verma ,International Journal of Engineering Research & Technology (IJERT) Vol. 2 Issue 11, IJERTISSN: 2278-0181, 11,November 2013, pp. 2099-2102.
20. "A minutiae based matching algorithm in fingerprint recognition systems": Łukasz Więcław, Journal Of Medical Informatics & Technologies Vol.13, ISSN 1642-6037,2009, pp.65-69.
21. "A tutorial on fingerprint recognition" : Davide Maltoni, ASB'03 Proceedings of the 1st international conference on Advanced Studies in Biometrics, June 02 - 06,2003, pp. 51-65.
22. "Fingerprint Matching Incorporating Ridge Features With Minutiae": H. Choi, K. Choi and J. Kim, IEEE Transactions on Information Forensics and Security, vol. 6, no. 2, 2011, pp. 338-345.
23. "A Fingerprint Recognizer Using Fuzzy Evolutionary Programming", T, V. Le, K. Y. Cheung and M. H. Nguyen, Hawaii International Conference on System Sciences, 2001, pp.1-5.
24. "A new fuzzy logic based image enhancement", M. Hanmandlu, S. N. Tandon, and A. H. Mir, Biomed. Sci. Instrum, vol. 34, 1997, pp. 590–595.
25. "A Survey of Thinning Methodologies", E. Hastings, Pattern analysis and Machine Intelligence, IEEE Transactions, vol. 4, Issue 9, 1992, pp. 869-885.
26. "A Multichannel Approach to Fingerprint Classification", A.K. Jain, S. Prabhakar, L.Hong, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 21, Issue 4, April 1999, pp. 348-359.

AUTHORS PROFILE

Sonali Sen received her M.Sc. degree in Computer Science from the University of Calcutta in the year 2003 and done her M.Tech in Computer Science and Engineering from the University of Calcutta in the year 2005. She is pursuing her Ph.D on Signal Processing from West Bengal University of Technology. She is working as Assistant Professor (Sr.) in the department of Computer Science, St. Xavier's College, Kolkata from 2006. She has published more than 15 journal and conference papers. Her research interests include Signal Processing, Graph Theory, Security, applied cryptography and Image Processing.

Deyashini Bhattacharya has received the B.Sc degree in Computer Science from St. Xavier's College (Autonomous), Kolkata and is currently pursuing her Master degree from NIT, Surathkal. She is passionate about programming. Her areas of interest include Discrete Mathematics, Data Structures and Algorithms, Data Security, Image Processing and Ethical Hacking. She is a technophile, shows interest to know the latest advancements and to find an alternative optimized solution by incorporating different concepts.



Soumili Dey has received the B.Sc degree in Computer Science from St. Xavier's College (Autonomous), Kolkata and is presently pursuing her master degree from the same college. She is proficient in programming languages like Java, C, C++. Her areas of interest include discrete mathematics, cryptography, web development and image processing. She likes researching new technologies and innovating old technologies. She tries knowing technologies from scratch which helps her in developing new ideas. She likes simplifying algorithms and utilizes the concept in her research work. Her knowledge in mathematics helps her create technical competency and firm mathematical foundation in her proposed algorithms.

Sabarna Nandy has received the B.Sc degree in Computer Science from St. Xavier's College (Autonomous), Kolkata and is presently employed in a reputed MNC. His areas of interest include Graph Theory, Data Security, Theory of Computation, image-processing and pattern recognition. His way of simplifying problems and solving programs basic concepts makes his algorithms and designs more efficient.