Author(s): Hossein Jahankhani
Article Title: Analysis of Fingerprint Image to Verify a Person
Year of publication: 2011
Citation: Jahankhani, H. & Mohid, M. (2010) 'Analysis of Fingerprint Image to Verify a Person', *Global Security, Safety, and Sustainability Conference*. (CCIS vol 92) Braga, Portugal, 1-3 September 2010. Springer, pp. 104-119.

Analysis of fingerprint image to verify a person

Hossein Jahankhani, Maktuba Mohid

School of Computing, IT and Engineering
University of East London, UK
h.Jahankhani@uel.ac.uk

Abstract: Identification and authentication technologies are increasing day by day to protect people and goods from crime and terrorism. This paper is aimed to discuss fingerprint technology in depth and analysis of fingerprint image. Verify a person with a highlight on fingerprint matching. Some fingerprint matching algorithms are analysed and compared. The outcomes of the analysis has identified some major issues or factors of fingerprinting, which are location, rotation, clipping, noise, non-linear distortion sensitiveness/ insensitiveness properties, computational cost and accuracy level of fingerprint matching algorithms. Also a new fingerprint matching algorithm proposed in this research work. The proposed algorithm has used Euclidean distance, angle difference, type as matching parameters instead of specific location parameter (like, x or y coordinates), which makes the algorithm location and rotation insensitive. The matching of local neighbourhoods at each stage makes the algorithm non-linear distortion insensitive.

Introduction:

Human bodies contain unique characteristics. Each of these characteristics differentiates one from another and therefore make easier to identify. The most traditional method for identification and authentication is the use of fingerprint technologies, [12]. This research project is bio-technology related project where biometric measurement is the main focus. The physiological biometrics such as fingerprints, iris and retinal scanners were developed in early stages. This research project based on fingerprinting process; especially fingerprint matching processes and related algorithms have been analysed, which leads to discover a new algorithm with some new features, [16] and [17].

A new algorithm has been proposed regarding those major factors where algorithmic solution has been given to overcome those major factors.

The target of this project is to find a standard, efficient and accurate fingerprint matching algorithm. To find efficient, standard and accurate fingerprint matching algorithm, first of all the factors/ issues by which the efficiency and accuracy of fingerprint matching algorithm can be measured need to be identified. The aim of this paper is to identified factors and issues by comparing and analysing different types of fingerprint matching algorithms. The factors/ issues are:

1. Rotation insensitiveness
2. Location insensitiveness
3. Clipping insensitiveness
4. Non-linear distortion insensitiveness
5. Noise insensitiveness
6. Low cost
7. High accuracy level

Background:

The word 'Bio' means biology and 'metric' means measure, so the combined word 'Biometric' means biological measurement. In general, biometric term is used relating to the identification of a person using biological characteristics with uniqueness property of human being. Two types of biological characteristics are used;

1. Behavioural – relating to the behaviour of a person. Like, voice recognition, typing rhythm etc. [14]
2. Physiological – relating to the shape, size or other properties of different parts of the body. It includes finger print recognition, face recognition, hand/ palm geometry recognition, DNA identification, retina-based measure etc.

All these types of biometric measures are almost unique for every person. Of them fingerprint recognition is the most popular one due to its immutability and uniqueness properties.

Immutability: Fingerprint pattern of a person does not change during his whole life other than the size and due to some special incidents, like accident, skin diseases, operation etc. This characteristic is called immutability.

Uniqueness: Fingerprint pattern is unique for every person, i.e. one person's fingerprint does not match with another person. This characteristic is called uniqueness. Even one person's fingerprints of two fingers do not match. Hand geometry, face recognition, behavioural biometric measures etc are not that much unique like fingerprint.

Also finger print image is easier to take than many other biometric measures. In case of retina-based measure, lighting is necessary.

Fingerprint patterns are normally categorized by friction ridges. The skin of inside surface of hands and fingers contain minute raised ridges known as 'friction ridge skin'. The minutiae of fingers can be categorized by some principal categories –

1. Ridge ending: the ending point of a ridge.
2. Bifurcation: The division of a ridge into two ridges.
3. Dot: Ridge like a dot, i.e. having equal width and length.
4. Spur: One type of bifurcation where a short ridge is a branch of a long ridge.

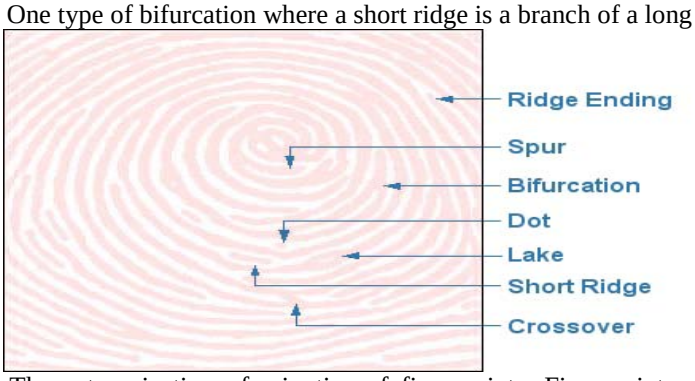

Image 1: The categorization of minutiae of fingerprints, Fingerprint analysis - the basics, [13].

5. Lake/ Enclosure: One ridge divides into two ridges and then reunites again.
6. Short ridge: A ridge which is short in length.
7. Bridge: A bridge-ridge that connects two ridges.

Combinations of these different types of ridges form a pattern of fingerprint. The positions and numbers of these different types of ridges create different types of fingerprint patterns, which show uniqueness.

Types of fingerprint patterns are available:

1. Loop: Ridges enter from one side, make curves and then go out from the same side. The centre of a loop is called core. The area of triangulation and division of ridges (bifurcation) is called delta. These two focal points, i.e. core and delta make a loop. In case of loop pattern, core, delta, area of delta are recorded, [15].
2. Whorl: Circular shaped ridges. Two or more deltas make a whorl. In case of whorl pattern, deltas and areas of deltas are recorded.
3. Arch: Ridges enter from one side, make a raise and then go out from the opposite side. Arch does not contain any core or delta. In case of arch pattern, whole area is recorded.

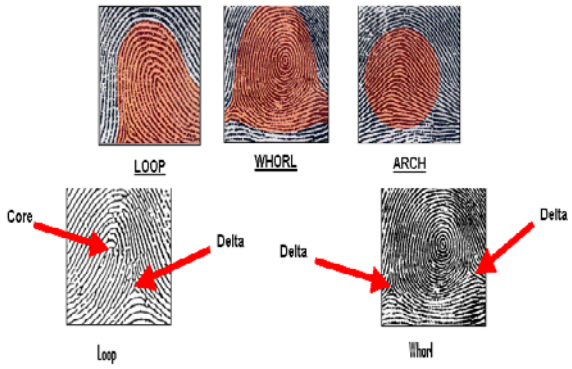

Image 2: Fingerprint patterns, legible fingerprints, Federal Bureau of Investigation.

The full process of fingerprint analysis is done using different steps:

1. Step 1: Image recording
2. Step 2: Image processing
3. Step 3: Image verification

Different types of fingerprint recording impressions are used:

Image recording is a very important part of fingerprinting. Image is taken over ink and paper, Porelon pad etc. Different types of devices or readers are used for recording fingerprint, like livescan devices, solid-state reader, optical reader, touch screen etc.

1. Rolled impression: In this case finger is rolled from one side of the finger nail to the other.
2. Plain impression: Finger is placed in 45° angle and fixed over the recorded material. It is used to verify the accuracy of rolled impression of fingerprint.

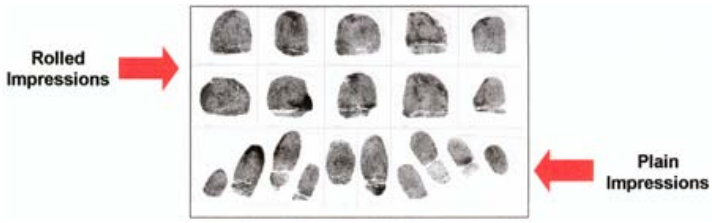

Image 3: Fingerprint recording, legible fingerprints, Federal Bureau of Investigation [11].

Image processing and image enhancement

Fingerprint is one of the most popular biometric measurement techniques. But the main problem arising regarding fingerprint is unclear image which is not suitable for image verification or might give wrong result. The image becomes unclear due to various reasons, like poor skin and ridge condition (skin disease, occupational mark etc), wrong way of taking fingerprint, wrong placement of fingers, poor quality of scanning device etc, [2], [4] and [5].

In this regards three types of error appears in fingerprint image:

False minutiae are created True minutiae are missed Position and orientation of minutiae are changed.

Image 4: Poor quality fingerprint images, [1].

Investigations showed that three types of fingerprint image regions occur:

Well-defined region: In which region different categories of minutiae (ridges, bifurcation, spur etc) are easily recognized and extracted using minutiae extraction algorithm, is called well-defined region. Recoverable corrupted region: In which region different categories of minutiae are not easily recognized, but can be extracted with the help of neighbouring regions, is called recoverable corrupted region.

Unrecoverable corrupted region: In some regions of fingerprint image, different categories of minutiae are not totally recognized (sometimes invisible) due to noise or distortion. Even the neighbouring regions become unable to provide enough information to extract minutiae patterns from those corrupted regions. This type of corrupted region is called unrecoverable corrupted region.

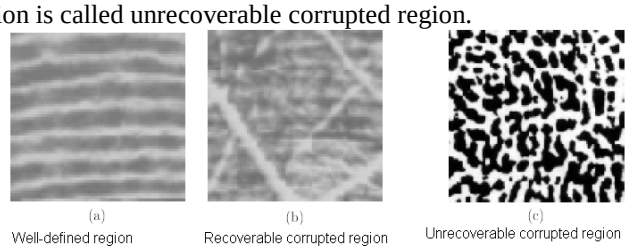

Image 5: Fingerprint regions, [1].

So after getting fingerprint image, it needs to be processed or enhanced to make it clear and suitable for identification. Two types of enhancement methods are normally used:

1) **Binarization-based method:** In this method, first of all image is transferred to binary image and then minutiae are extracted from the binary image.

2) Direct gray-scale enhancement process: Minutiae can be extracted from gray-scale image directly without transforming into binary image using a direct gray-scale enhancement algorithm.

Fingerprint matching

Fingerprint matching is one of the oldest, reliable and widely used biometric measures. Normally two approaches are used in case of fingerprint matching:

1) Minutiae method: In minutiae method, the information of ending (terminal point) and bifurcation (separation point) of ridges, like type of ridges, position, direction etc are matched.

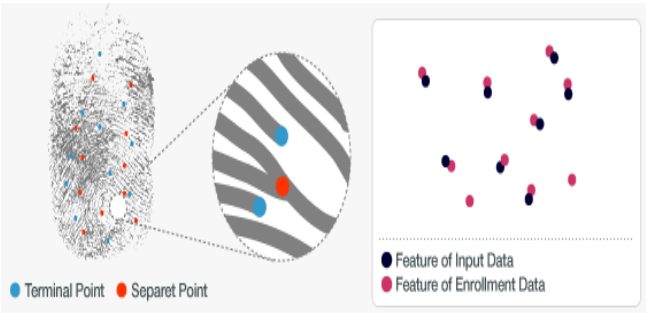

Image 6: Minutiae method, fingerprint Verification Algorithm, [10].

2) Pattern matching method: In pattern matching method, the input fingerprint image and registered fingerprint image are compared using direct comparison of flow of ridges of all places of the images.

Image 7: Pattern matching method, fingerprint Verification Algorithm, [10].

Image transformation:

Sometime enrolee and claimant fingerprints are not in a same format, like rotated in different angles, different scaling, different sizes etc. In those cases, fingerprint images

need to be transformed to make the formats same. Otherwise, feature extraction, matching might be wrong, which might give wrong result. Different types of transformations are used:

1) Rigid transformation:

Rotation, shifting etc are called rigid transformation.

2) Affine transformation: Scaling, shearing etc are called affine transformation.
3) Non-linear transformation:

Sometimes some transformations need to be applied on the image with some complex equations. This type of transformation is applied using thin plate spline deformation model or by modelling as piece-wise linear transformation, [8], [18].

Fingerprint verification:

In this case, first of all fingerprint of enrolee is recorded and put in the database with his/her identity. Later on, when anyone claims to be that person, the claimant fingerprint is matched with enrolee one for verification. It is used for accessing secured systems, entering secured areas etc. As one fingerprint is matched with another fingerprint in the database, it is called "one to one matching".

Different approaches or methods are used for fingerprint verification, which are –

1) Neighbourhood minutiae comparison
2) Modified approach of neighbourhood comparison.
3) Global matching/correlation approach
4) Modified correlation approaches

Acceptance/ rejection results, i.e. recognition rate, accuracy, as well as cost of fingerprint verification are dependent on the threshold value. There are two types of recognition rate: false acceptance rate (FAR)/ false match rate/ Type II error and false rejection rate (FRR)/ false non-match rate/ Type I error. If the threshold value is chosen to be very high, number of false rejections (FRR) will be increased. In this case, it might happen that fingerprints of same finger of same person, which should be matched, will not be matched due to noisy image. On the other side, if the threshold value is chosen to be very low, number of false acceptance (FAR) will be increased. In this case, it might match two fingerprints which should not be matched. So threshold value should be chosen in a range so that false rejection and false acceptance will not occur.

There are two types of ways of choosing threshold value:

1) Back-end adjustment
2) Front-end adjustment

Fingerprint identification:

Fingerprint identification is used to identify criminals, where fingerprint left by criminal is matched with many fingerprints of database. As one fingerprint is matched with many unknown fingerprints, it is called "one to many matching".

In case of fingerprint identification, claimant fingerprint is matched with a large number (10 to ten million) of enrolee fingerprints. It will be time consuming if fingerprint verification process is applied to match each of these claimant-enrolee fingerprint pairs. That's why, in case of fingerprint identification, different approach is used, which is a two step process.

- **Step 1:** Test Fingerprint and fingerprints of database are categorized into different types of pattern classes (whorl, loop, arch etc). After classification process, pattern matching method is used for the comparison of test fingerprint and fingerprints of database. Step 1 is a fast process.
- **Step 2:** Minutiae method is applied over all of those pairs of claimant-enrolee fingerprints which show little difference in step 1 results. This step is time consuming.

Different types of fingerprint matching algorithms

There are different types of minutiae based fingerprint matching algorithms are available. Three of them have been discussed, analysed and compared here. They are graph-based fingerprint matching, fingerprint matching using onion layer algorithm of computational geometry and K-plot algorithm, [9].

Fingerprint matching using graph matching approach: The graph based fingerprint matching algorithm which has been discussed here is proposed by D.K. Isenor and S.G.Zaky (Department of Electrical Engineering, University of Toronto, Canada). The algorithm converts the whole image of fingerprint into graphs where ridges are considered to be nodes of graphs and relationships between ridges are considered to be the connections of nodes.

Following information is kept for each node/ ridge:

a) The ridge length;
b) Nature of minutia in each end (i.e. ending or bifurcation);
c) If any of the minutiae endings is bifurcation, the number of end neighbours involved in that end;
d) The information whether the ridge is complete or not;
e) The list of ridge neighbours ordering by the clockwise direction starting from end 1;
f) The overlap length for each neighbour.

After converting images into graphs, the graphs are matched using those information (given above) of ridges/ nodes. The noise reduction step makes the algorithm insensitive to noise, displacement and distortion.

Some steps are followed for graph based matching. The steps are given below:

- Step 1 – Ordering of sides and endings;
- Step 2 – Identifying neighbours;
- Step 3 – Numbering the levels;
- Step 4 – Drawing the graph;
- Step 5 – Reducing noise;
- Step 6 – Partitioning;
- Step 7 – Refinement;
- Step 8 – Scoring, [6].

Fingerprint matching using onion layer algorithm of computational geometry:

This fingerprint matching approach has been proposed by [7]. The algorithm is based on onion layer algorithm of computational geometry. The algorithm generates nested

polygons using repeated convex hull algorithm where nodes of the polygons are the minutiae points of fingerprint image. Thus it is minutiae based algorithm. The whole algorithm can be divided into three parts:

a) Finding nested polygons using convex hull algorithm and assigning depth number.

b) Finding reference triangle for calculating rigid transformation parameters and local matching

c) Pair matching (global matching) where it compares type, x, y coordinates and orientation angle of minutiae points.

This algorithm rejects unmatched fingerprints before global matching, which saves time by avoiding time consuming global matching steps.

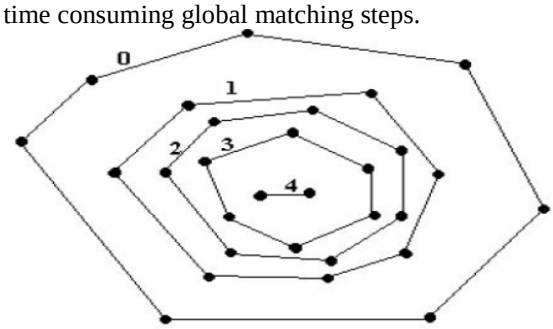

Image 8: Nested polygons constructed from point set, [7].

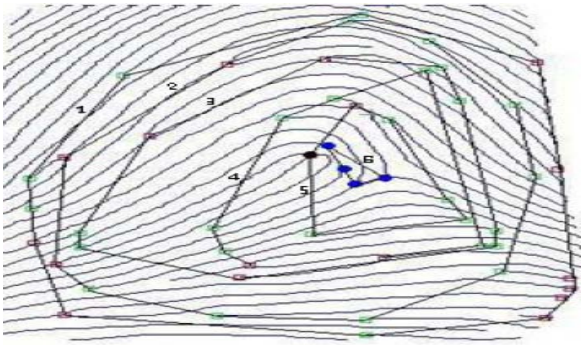

Image 9: Fingerprint with minutiae and its nested polygons, [7].

Fingerprint matching using K-plet and Coupled BFS algorithm:

This algorithm has been proposed by [3], (Centre for Unified Biometrics and Sensors, University at Buffalo, NY, USA). The algorithm is based on graph matching principles and is divided into 2 parts:

1. Graph representation named K-plet.
2. Matching using coupled BFS (Breadth First Search) algorithm.

A graph is drawn using K nearest neighbours of all minutiae points. After extracting minutiae points, this algorithm draws a graph using K nearest neighbours of all minutiae points where each neighbour contains following data --

Euclidean distance, orientation angle difference, connecting edge direction between specific minutia point and the neighbour.

There are two approaches of finding K nearest neighbours:

1. K-nearest neighbours depending on Euclidian distances.
2. Nearest neighbour in each quadrant sequentially depending on Euclidian distances.

After drawing graphs, they are traversed and matched using coupled BFS (CBFS) algorithm. The CBFS algorithm is a dual graph traversing algorithm, i.e. it traverses both template and input fingerprint graphs at the same time and matches pair of minutiae points using their neighbour data. Two minutiae points are said to be matched only if all of their neighbours are matched.

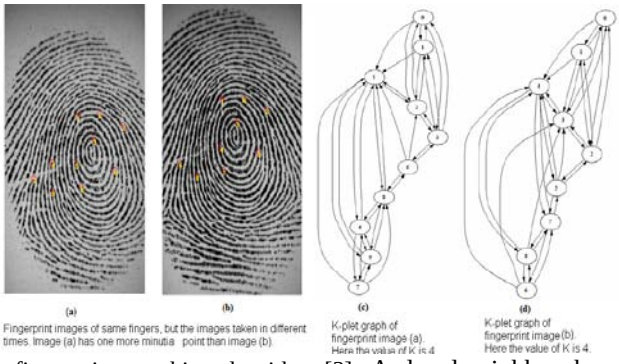

Image 10: K-plet fingerprint matching algorithm, [3]. As local neighbourhoods are matched at each stage, the algorithm is insensitive to non-linear distortion.

No explicit alignment is required in this algorithm as distance and angle differences are used instead of particular location parameters (like, x,y coordinates).

Experimental procedure:

The algorithm which has been proposed in this thesis is minutiae based algorithm, which matches all the neighbours of all minutiae points.

The minutiae points are said to be matched if type and all neighbours are matched. Two neighbours of two minutiae points are said to be matched if type, Euclidian distance, angle difference are matched. The minutiae points and their neighbours are matched recursively.

The neighbours of any minutia point are sorted according to their Euclidian distances in increasing order. And one additional feature is added in case of matching neighbours of minutiae points, which decreases the cost of matching.

The matching runs for all neighbours of all minutiae points of enrolee/template fingerprint against all neighbours of all minutiae points of claimant/input fingerprint. It stops matching if Euclidian distance of claimant/input neighbour becomes greater than Euclidian distance of enrolee/template neighbour. As the neighbours of any minutia point are sorted according to Euclidian distances in increasing order, if any Euclidian distance of any claimant/input neighbour becomes higher than Euclidian distance of enrolee/template neighbour, the Euclidian distances of later neighbours of claimant minutia point will be more higher than the current one and it will not match, so there is no point to continue matching, thus in this way it decreases some cost of matching. The feature has been clearly described in pseudo code and Matlab code.

Another modification has also been proposed, which will deal with the problem of clipping image. If any image is clipped from any side, the minutiae points of the

discarded side will not be found and remain unmatched, which might affect the matching result. The proposed algorithm detects those minutiae points which remain in the discarded side and then it subtracts the numbers from the result, so that it might not affect the result.

At the time of matching the neighbours of two minutiae points, if any neighbour of any fingerprint image remains unmatched, the proposed algorithm investigates whether the x or y coordinate of that neighbour is greater or smaller than the x or y coordinates of all the minutiae points of another fingerprint image. That means it investigates whether the neighbour minutia point remains in the boundary of any side. Because clipping of image occurs in boundaries. If the point remains in the boundary, it then checks whether there is any other neighbour of the same image, which remains in the same side of boundary after that unmatched neighbour and which is matched. If any neighbour exists like that, the unmatched neighbour is not considered to be in the discarded portion of cropped image. Otherwise, it is considered to be existed in the discarded portion of cropped image and then it is not considered or counted in calculation of matching score.

After doing matching, if any minutia point of any image remains unmatched, the algorithm does the same thing, i.e. investigates whether the minutia point remains in any side of the discarded portion of cropped image or not. If the minutia point remains in the discarded portion of cropped image, the algorithm does not count it in the calculation of final matching score. For matching purpose, this suggested algorithm use Euclidian distance, angle difference and type instead of using particular location parameters (like x, y coordinates), which makes the algorithm insensitive to location and rotation.

As local neighbourhoods are matched at each stage, the algorithm is insensitive to non-linear distortion as well.

All these features make this suggested algorithm insensitive to location, rotation, non-linear distortion and clipping. No explicit alignment is needed due to these insensitiveness properties, which decreases cost as a whole.

The algorithm has been described below:

After extracting minutiae points, each minutia point has been assigned a unique ID and status. At first the statuses of all minutiae points have been set to unmatched. Then the neighbour data has been calculated for all minutiae points for both of the two images. The neighbour data contains neighbour minutia point ID, matching status, Euclidian distance and orientation angle difference between the neighbour and that minutia point. At first the matching statuses of all neighbours have been set to unmatched. The neighbours of any minutia point have been sorted according to their Euclidian distances in increasing order.

After gathering, calculating and collecting all data/information, the minutiae points of two images have been matched using Pair_Matching function (which will be described later) if the minutiae points do not have matched status value. After matching all minutiae points of two images, the total number of unmatched boundary points of two images, which remain in the discarded portion of cropped image, has been calculated using checkBoundaryPoint function (which will be discussed later). This function checks whether the unmatched point remains in the discarded portion of cropped

image or not. If it remains in any side of discarded portion, it will not be counted in calculation of final result.
After this boundary point checking, the number of matched pairs has been calculated using statuses of minutiae points of enrolee image.
At last the matching score or result has been calculated using following equation:

$$\text{Result} = \frac{2 \times \text{Number of matched pairs of two fingerprints}}{\text{number of minutiae points of input fingerprint} + \text{number of minutiae points of template fingerprint} - \text{Total number of unmatched boundary minutiae points of two fingerprints, which remain in the discarded potion of clipped image}} \times 100$$

Here, 100 are multiplied to get a percentage value of the matching score. If the result is greater than or equal to the chosen threshold value, the two images will be said to be matched, otherwise, they will be said to be unmatched. The pseudo code and Matlab code of this function are described in appendix 3 and 4 correspondingly.
The Pair_Matching function is used for matching of two minutiae points of two fingerprints. The function proceeds only if status of none of the minutiae points is matched. In this function, a matching score has been calculated for the pair, which has been initially set to 0 values. Nested loops are present in this function. The outside loop is for the neighbours of enrolee/template minutia point and inner loop is for the neighbours of claimant/input minutia point. Inside two loops, two neighbours have been checked for matching. Two neighbours will be said to be matched if---

$$\textbf{Result} = \begin{cases} yes, if \quad |Dist_i - Dist_t| < r_0 \\ Type_i = Type_t \ and \\ |\theta_i - \theta_t| < \theta_0 \\ no, Otherwise \end{cases}$$

Here $Dist_i$ is the Euclidian distance between the neighbour and minutia point of input image, $Dist_t$ is the Euclidian distance between the neighbour and minutia point of template image, $Type_i$ is the type of the neighbour of input minutia point, $Type_t$ is the type of the neighbour of template minutia point, θ_i is the angle difference between the neighbour and minutia point of input image, θ_t is the angle difference between the neighbour and minutia point of template image. r_0 is the distance threshold, θ_0 is the angle threshold. That means two neighbours are said to be matched only if—

1. Their types are same;
2. Their Euclidean distance difference is lower than the distance threshold value;
3. The difference between their orientation angle differences is lower than the angle threshold value.

If the matching result comes yes, the matching statuses of both neighbours will become matched and matching score will be increased by one. If two neighbour minutiae points are matched and none of them has status (minutia point status) as matched, the Pair_Matching function will be called recursively using two neighbours as two minutiae points. If the Euclidian distance of current neighbour of input/claimant minutia point becomes higher than the threshold value plus the Euclidian distance of

the current neighbour of template/enrolee minutia point, the code will exit from the inner loop, which saves cost of matching.
After completion of outer loop, all unmatched neighbours of both of the input and template minutiae points have been checked to be the boundary points of the discarded portion of cropped image using checkBoundaryPoint function.
After that, a matching result has been calculated for the minutia point pair using following equation:

$$\text{Result} = \frac{2 \times \text{Total matching score of the minutiae point pair}}{\text{number of neighbours of input minutia point} + \text{number of neighbours of template minutia point} - \text{total number of unmatched boundary neighbours of two minutiae points, which remain in the discarded portion of clipped image}} \times 100$$

Here, a value 100 is multiplied to get a percentage of the result. If the result becomes greater or equal to the threshold value, two minutiae points will be said to be matched, otherwise, they will be said to be unmatched. If two minutiae points are matched, matched as status values will be assigned to them. checkBoundaryPoint function is used to check whether any unmatched minutia point of any fingerprint image exists in the discarded portion of another clipped fingerprint image. It checks whether x or y coordinate of the specific minutia point is greater or smaller than corresponding x or y coordinates of all minutiae points of another image. If x or y coordinate of the point is greater or smaller than corresponding x or y coordinates of all minutiae points of another image and if there exists no matched minutia point in the same portion of the same image after that unmatched minutia point, it will consider the point to be placed in the discarded part of the clipped image, and will get the reason of being unmatched.
The checkBoundaryPoint function works for each unmatched point of each image at a time. Nine steps are used in this case where eight steps are used for checking. The steps are:
1. Checks whether the x coordinate of the specific point of that image is greater than the x coordinates of all minutiae points of the counter image.
2. If the result of step 1 is true, it checks whether there exists any minutia point in the same image, whose status is matched and the x coordinate is greater or equals to the x coordinate of the specific point. If any point is found in this step, it passes to step 3, otherwise passes to step 9.
3. Checks whether the x coordinate of the specific point of that image is smaller than the x coordinates of all minutiae points of the counter image.
4. If the result of step 3 is true, it checks whether there exists any minutia point in the same image, whose status is matched and the x coordinate is smaller or equals to the x coordinate of the specific point. If any point is found it passes to step 5, otherwise passes to step 9.
5. Checks whether the y coordinate of the specific point of that image is greater than the y coordinates of all minutiae points of the counter image.
6. If the result of step 5 is true, it checks whether there exists any minutia point in the same image, whose status is matched and the y coordinate is greater or equals to the y

coordinate of the specific point. If any point is found in this step, it passes to step 7, otherwise passes to step 9.

7. Checks whether the y coordinate of the specific point of that image is smaller than the y coordinates of all minutiae points of the counter image.

8. If the result of step 7 is true, it checks whether there exists any minutia point in the same image, whose status is matched and the y coordinate is smaller or equals to the y coordinate of the specific point. If any point is not found in this step, it passes to step 9

9. It considers the specific point to be existed in the discarded portion of clipped image. This boundary point checking is done both for unmatched neighbours and unmatched minutiae points of both template and input fingerprint images. This function solves the problem of clipped/cropped image.

Cost:

The full matching algorithm is divided into 3 parts:

a) Sorting of neighbours for all minutiae points of two images
b) Recursive BFS algorithm for pair matching
c) Boundary point checking for unmatched neighbours and minutiae points of both of the images

Some sorting algorithms take linear time to run, i.e. cost is O(M), where M is the number of minutiae points. If the sorting of neighbours is done for all the minutiae points, the cost will be $O(M^2)$. As it is done for two fingerprint images, total cost will be $O(2M^2)=O(M^2)$.

The cost of BFS (Breadth first search) algorithm is O(|M|+|N|), where M is the number of minutiae points and N is the number of neighbours. As all other minutiae points are considered to be the neighbour, |N|=|M|-1. Thus the cost of CBFS/ recursive pair matching algorithm will be O(|M|+|M|-1)=O(M). As neighbours are sorted according to distances and loop does not continue matching if the distance of enrolee/template neighbour becomes less than that of claimant/input neighbour, the cost will be less than O(M). The boundary point checking is done in at most eight checking steps. So the cost of the boundary point checking of every unmatched neighbour or minutia point is O(8M)=O(M), where M is the number of minutiae points. If number of unmatched neighbours or minutiae points is C, the total cost will be O(CM)=O(M). So the total cost of matching will be $O(M^2)+O(M)+O(M)=O(M^2)$.

Analysis:

The three matching algorithms have been discussed, which are graph-based algorithm, K-plet algorithm, onion layer algorithm using computational geometry. Two algorithms from the three have been implemented, which are K-plet algorithm and computational geometry-based algorithm. After implementation, the three algorithms are analysed and compared.

Conclusion:

The main focus of this research work was fingerprint matching. There are different types of fingerprint matching processes or algorithms. Of them, some are discussed, implemented, analysed and compared. The main target of this thesis is to find an algorithm which increases performance and accuracy levels. After investigating a lot,

the following key points have been gathered which deal with the accuracy and performance issues of any fingerprint matching process:
Elastic distortion, location, rotation, clipping, noise insensitiveness and running cost. The computational geometry-based algorithm is non-linear distortion, location, rotation, clipping sensitive. For these sensitiveness properties, this algorithm needs to generate reference triangle to get transformation parameters, which increases the cost of the algorithm. Also, some data (type, orientation angle of minutia point and edges) is stored twice, which also increases storage cost.
The graph-based approach is noise insensitive. But for reducing noise, it uses the refinement process, which costs higher. The algorithm is rotation and location insensitive, but it is clipping and non-linear distortion sensitive.
The K-plet algorithm is non-linear distortion, location, rotation insensitive, but it is clipping sensitive. Furthermore chosen value of K affects the cost and accuracy of the algorithm. If higher value of K is chosen, the accuracy level becomes high, but cost will increase as well. On the other side, if lower value of K is chosen, the cost becomes low, but accuracy level also becomes low.
Thus, after the investigation, in the conclusion, it can be said that the following properties should be highlighted while choosing/selecting any fingerprint matching algorithm:

- Rotation insensitiveness
- Location insensitiveness
- Clipping insensitiveness
- Non-linear distortion insensitiveness
- Noise insensitiveness
- Low cost
- High accuracy level

That means chosen algorithm needs to have high accuracy level and low cost.
For increasing accuracy level, either the algorithm needs to be rotation, location, clipping, non-linear distortion insensitive or needs some transformation processes. Noise is another factor which deals with accuracy issue. Noise reduction is highly dependent on image enhancement processes of image enhancement stage. Noise can also be reduced by some refinement processes while matching.

References:

[1] Hong, L., Wan, Y., Jain, A., Fingerprint image enhancement: algorithm and performance evaluation. [Internet]. 20(8), pp. 777-789. Available from: http://www.cs.ru.ac.za/courses/Honours/mmcourse/security/biometrics/MSU-CPS-97-35.pdf [Accessed 4 July 2009] ,(1998)
[2] Greenberg, S., Aladjem, M., Kogan, D., (2002) Fingerprint image enhancement using filtering techniques, 8(3), pp. 227-236. [Internet]. Available from: http://www.cs.kent.edu/~wcheng/Presentation%201%20(Image%20Enhancement)/Fingerprint%20Image%20Enhancement%20using%20FilterTechniques.pdf [5 July 2009]
[3] Chikkerur, S., Cartwright, A.N., Govindaraju, V., (2006).K-plet and CBFS: A Graph Based Fingerprint Representation and Matching Algorithm. [Internet]. Available from: <http://web.mit.edu/sharat/www/research/papers/kplet_long.pdf>[21 July 2009]

[4] Kaur, M., Singh, M., Girdhar, A., Sandhu, P.S. (2008). Fingerprint Verification System using Minutiae Extraction Technique. [Internet]., Available from: http://www.waset.org/journals/waset/v46/v46-85.pdf [Accessed 23 July 2009]

[5] O'Gorman, L. Fingerprint Verification. (1999). [Internet]. (479), pp. 43-64. Available from: http://www.cse.msu.edu/~cse891/Sect601/textbook/2.pdf [Accessed 25 July 2009]

[6] Isenor, D.K., Zaky, S.G., (1986). Fingerprint Identification using Graph Matching*, 19(2), pp. 113 122.

[7] Khazaei, H., Mohades, A., (2007). Fingerprint Matching and Classification using an Onion Layer algorithm of Computational Geometry, [Internet]. 1(1). Available from: http://www.naun.org/journals/mcs/mcs-5.pdf [Accessed 26 July 2009]

[8] Chikkerur, S., Wu, C., Govindaraju, V.A (2004). Systematic Approach for Feature Extraction in Fingerprint Images. [Internet]. 3072, pp. 344-350. Available from: <http://www.cubs.buffalo.edu/pdf/finger-systematic.pdf>[15 September 2009]

[9] Corman, T. H., Leiserson, C. E., Rivest, R. L., (1990), Introduction to Algorithms, Cambridge, MA, U.S.A, MIT Press.

[10] DDS (Digital development System). (2006). Fingerprint Verification Algorithm. [Internet]. Available from: http://www.dds.co.jp/en/fv/algorithm.html [Accessed 21 July 2009]

[11] Federal Bureau of Investigation. Taking legible fingerprints. [Internet] Available from: http://www.fbi.gov/hq/cjisd/takingfps.html [Accessed 20 June 2009].

[12] Grannblog. Problems with fingerprints for authentication. [Internet] Available from: http://blog.granneman.com/2006/05/13/problems-with-fingerprints-for-authentication/ [Accessed 20 June 2009].

[13] Crimtrac. (2008). Fingerprint analysis – the basics. [Internet] Available from: <http://www.crimtrac.gov.au/systems_projects/FingerprintAnalysis-TheBasics.html> [Accessed 20 June 2009].

[14] IT University Copenhagen. Morphology. [Internet]. Available from: http://www.itu.dk/courses/MBSB/E2005/Download/morphology.pdf [5 July 2009]

[15] Atalasoft DotImage 8.0, (2009). Morphology. [Internet]. Available from: <http://www.atalasoft.com/Products/dotimage/docs/Morphology.html> [5 July 2009]

[16] University of Wales Aberystwyth. (2007). Aberystwyth Users site. Available from: http://users.aber.ac.uk/sac4/cs180/index.html [Accessed 20 June 2009]

[17] Wikipedia. (2009). Biometrics. [Internet] Available from: <http://en.wikipedia.org/wiki/Biometrics> [Accessed 20 June 2009].

[18] Wikipedia, (2009). Transformation matrix. [Internet]. Available from: http://en.wikipedia.org/wiki/Transformation_matrix [Accessed 22 July 2009]