CS698x

Project Report

Finger Print matching using singular point

Submitted by:
Akarshan Sarkar
Arpit Jain

Project Mentor:
Prof. Phalguni Gupta
Dr. Kamlesh Tiwari

Abstract

The study focused on performing fingerprint matching between fingerprints for the same person. We used the information of core points, delta points and the minutiae points in doing the matching. Obtaining the point of alignment between two sets of minutiae is a combinatorial problem and is computationally expensive. In this project, we study how the detection of singular points (core and delta points) helps in matching by giving a similarity score (or dissimilarity score) between two fingerprints of the same person in fingerprint images by reducing the complexity and improving the performance of the matcher.

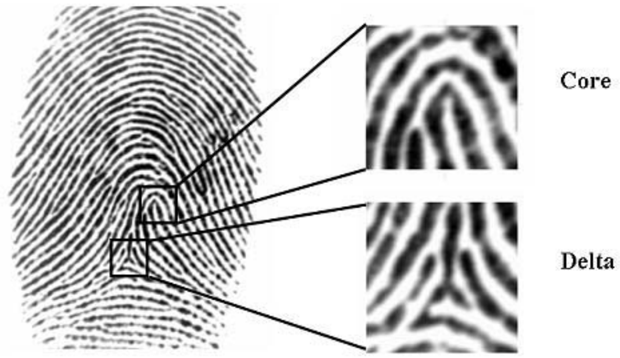
1 Introduction

Fingerprint recognition is a well researched topic and a large number of algorithms have been proposed to date in the literature. Many algorithms exist like image correlation, filterbanks or minutiae points and texture descriptor etc [3]. Among these minutiae based algorithms are most widely used which depend on explicit or imlicit alignment of minutiae for matching the two fingerprint images. Fingerprint acquisition causes to place the fingerprints randomly to an extent which makes finding point correspondences and hence the process of alignment a combinatorial one. Using singular points as prior knowledge will lead us to solve the problem of matching efficiently with almost negligible drop in the accuracy.

2 Related work

In this paper [1], a new representation for encoding the local neighborhood of each minutia called K-plet has been introduced from which we derive our work.

Singular points (or the global features of the fingerprints such as the core and delta) are the landmark points which remains almost consistent across various other impressions of the same user. These are special pattern of ridge and valleys. Like for example, a core is defined as the top most point on the inner most ridge. A delta is defined as the center point where three ridges with different direction flows meet.



We used the information from singular points to reduce the number of runs for each possible cases giving a drop in time computation.

3 Work done

We were given a database of 500 subjects with two images for each subject. We were also given the position of marked minutiae points(position and orientation). Our task was just to build the matcher.

For finding the singular points, we used a marker tool in matlab to manually mark the singular points in 1000 images. Although this is prone to some errors but we will see our algorithm doesn't depend much on the exact position of the singular points.

A figure showing minutiae points (green in color) and singular points (core point only in this example shown in red color) from the images of the dataset given to us.

Methodology

K-plet representation consists of a minutiae m_i and K other minutiae $(m_1, m_2, m_3, ...m_K)$ taken from its local neighborhood. This whole structural relationship of K-plet is encoded in the form of a graph $G(V, E)$ where each minutiae is a vertex v and each neighboring minutiae is represented by a directed edge (u, v). Each vertex or minutiae is attributed by (x_v, y_v, θ_v) which represents the coordinate and orientation of the minutiae respectively. Each directed edge (u, v) is represented by co-ordinates $(r_{uv}, \phi_{uv}, \theta_{uv})$ which represent Euclidean distance between minutiae m_a and m_b, direction of edge connecting the two minutiae and the relative orientation of minutiae m_j w.r.t the central minutiae m_i respectively.

In the matching algorithm, we match local neighborhood of minutiae points in the K-plet representation. For this the neighborhood of minutiae points are ordered in the increasing order of the radial distance r_{ij} which then reduces tothe problem of matching two ordered sequences. We applied a dynamic programming approach based on string alignment algorithm discussed in [2]. We initially used a greedy approach but it was too much time consuming.

For matching two fingerprint images, we obtain the K-plet graphs G and H respectively from their minutiae points. Starting from source points in both the graphs G and H, we traverse the graph in BFS fashion finding the number of minutiae pairs matched. This step is consolidated by choosing only those vertices in the adjacency list of the each node in the graph G and H which are themselves locally matched.

So, we do this for all possible pairs of source nodes in the two graphs and find the number of maximum minutiae pairs matched - m. We then decide the similarity score between the fingerprint images based on this number m, normalised by M_R, M_T which represents the number of minutiae points in the reference and template image respectively.

$$s = \left\{ \begin{array}{l} \frac{m^2}{M_R M_T}, \text{ if, } M_R \geq M_{Th}, M_R \geq M_{Th} \\ max\left(\frac{m}{M_R}, \frac{m}{M_T}\right), \text{ otherwise} \end{array} \right\}$$

M_{th} is a threshold on the number of minutiae (we used 15 in our experments). This is useful for cases where number of minutiae in one print is very low and in other is high.

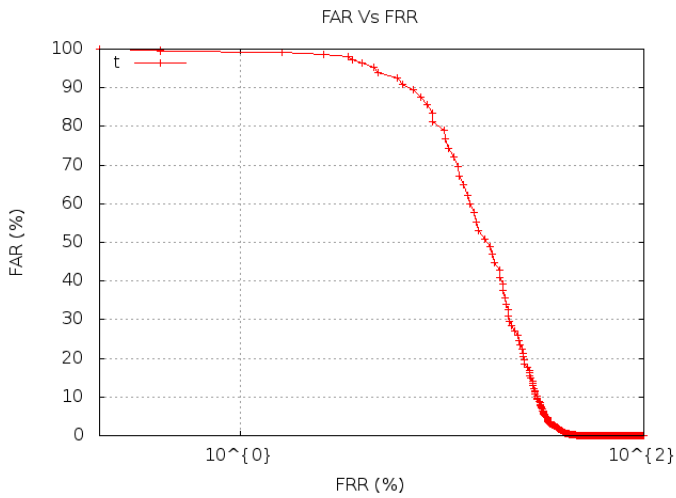
Note that this K-plet technique is rotation and translation invariant since it defines its own local coordinate system and we did our matching with those coordinate system. Therefore registering the fingerprint images using the singular points will be of no use. This method is even capable of matching when one of the print is rotated by 180 degrees.

But given the singular points we can constrain the set of possible source nodes in both fingerprint images. We chose to consider only c minutiae points around singular points, so this approach will lead to c^2 number of repetitions of BFS travesal with different source nodes. Had it been all possible source node pairs, it would be n^2 where n is number of minutiae in both the images. In our experiments, we took c to be 15. Note that, we used only the c nearest minutiae points around the singular point in both images. So having some error in position of the singular points won't affect the overall accuracy.

4 Results

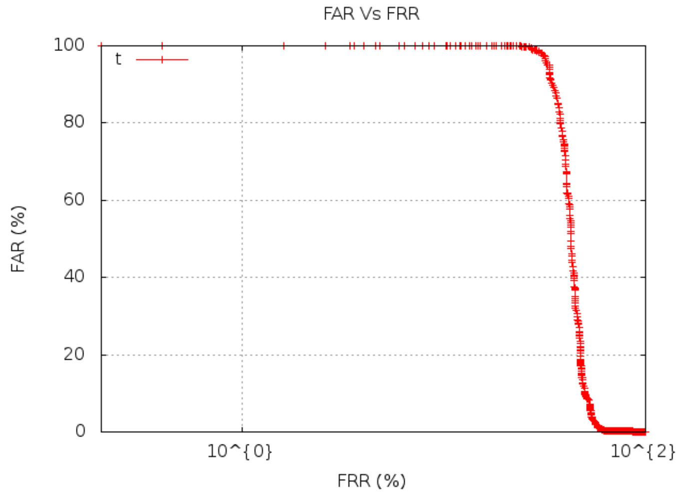
K represents the number of neighborhood vertices chosen for each minutiae in the K-plet representation.

K=4



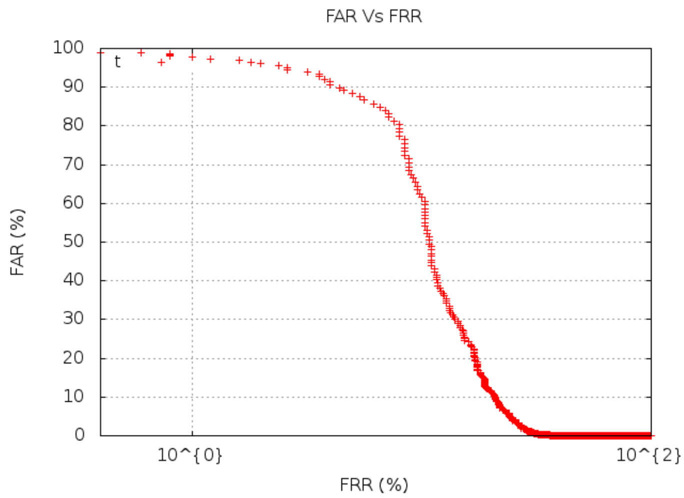
FAR vs. FRR report:

For best case:

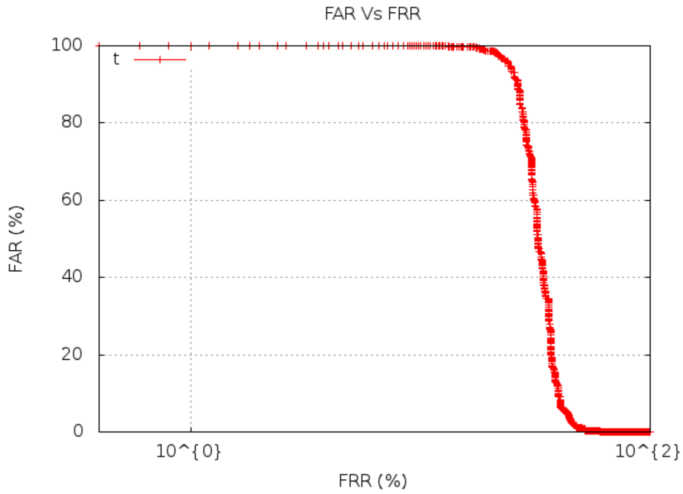


FAR vs. FRR report:
Correct Recognition Rate (CRR) = 56.6
Equal Error Rate (EER) = (24.310220) 0.994300 with Difference = 0.620441
False Acceptance Rate (FAR) = ∼ 24.058
False Rejection Rate (FAR) = 24.2
Total No. of Falsely Reject Matching = 121 out of total 500 genuine matching

K=8

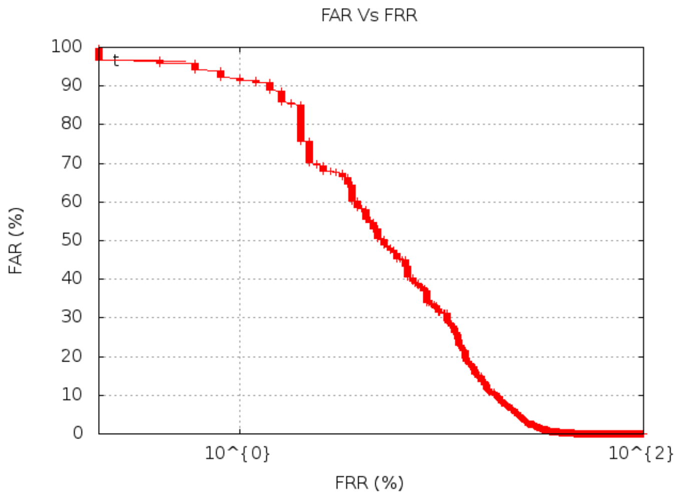


FAR vs. FRR report:
For best case:

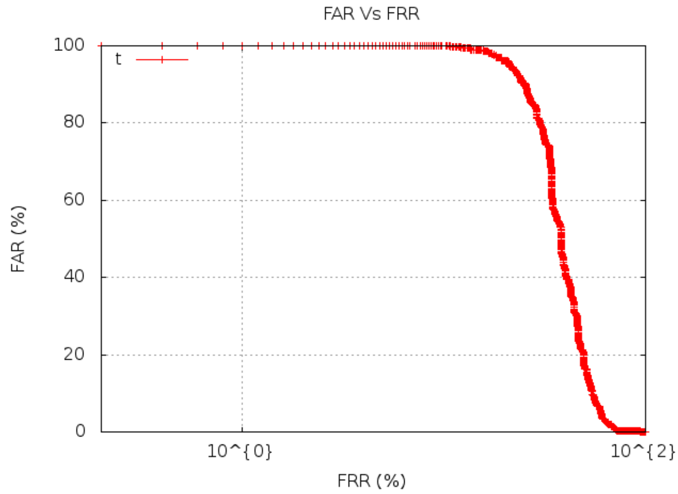


FAR vs. FRR report:
Correct Recognition Rate (CRR) = 68
Equal Error Rate (EER) = (17.311823) 0.984600 with Difference = 0.176353
False Acceptance Rate (FAR) = ∼ 17.08456913827655310621
False Rejection Rate (FAR) = 17.4
Total No. of Falsely Reject Matching = 87 out of total 500 genuine matching

K=16



FAR vs. FRR report:
For best case:



FAR vs. FRR report:
Correct Recognition Rate (CRR) = 61.2
Equal Error Rate (EER) = (15.197595) 0.892800 with Difference = 0.004810
False Acceptance Rate (FAR) = $\sim$ 15.2
False Rejection Rate (FAR) = 15.2
Total No. of Falsely Reject Matching = 76 out of total 500 genuine matching

5 Future Work and conclusion

In this academic project, we made a fingerprint minutiae based matcher. We used a popular K-plet technique for matching which incorporates singular points in the fingerprint images and matches the two fingerprint images more efficiently. We detected the singular points manually which can be automated by using complex filtering techniques which remains to be our future work.

References

[1] Sharat Chikkerur, Alexander N. Cartwright, and Venu Govindaraju. K-plet and Coupled BFS: A Graph Based Fingerprint Representation and Matching Algorithm. In David Zhang and Anil K. Jain, editors, *ICB*, volume 3832 of *Lecture Notes in Computer Science*, page 309315. Springer, 2006.

[2] Thomas H. Cormen, Clifford Stein, Ronald L. Rivest, and Charles E. Leiserson. *Introduction to Algorithms.* McGraw-Hill Higher Education, 2nd edition, 2001.

[3] Anil K. Jain and David Maltoni. *Handbook of Fingerprint Recognition.* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003.