Fingerprint Recognition

Andrew Ackerman
Professor Rafail Ostrovsky

Abstract

Fingerprint matching is the process used to determine whether two sets of fingerprint ridge detail come from the same finger. There exist multiple algorithms that do fingerprint matching in many different ways. Some methods involve matching minutiae points between the two images, while others look for similarities in the bigger structure of the fingerprint.

In this project we propose a method for fingerprint matching based on minutiae matching. However, unlike conventional minutiae matching algorithms our algorithm also takes into account region and line structures that exist between minutiae pairs. This allows for more structural information of the fingerprint to be accounted for thus resulting in stronger certainty of matching minutiae. Also, since most of the region analysis is preprocessed it does not make the algorithm slower.

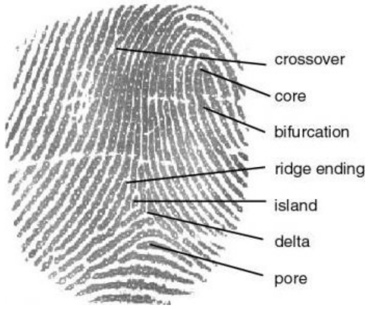
The algorithm for matching was not created; however, the process in which the regional data is obtained is explained in this paper. Evidence from the testing of the preprocessed images gives stronger assurance that using such data could lead to faster and stronger matches.

Introduction

Extensive research has been done on fingerprints in humans. Two of the fundamentally important conclusions that have risen from research are: (1) a person's fingerprint will not naturally change structure after about one year after birth and (2) the fingerprints of individuals are unique. Even the fingerprints in twins are not the same. In practice two humans with the same fingerprint have never been found.

The Basics

A fingerprint is comprised of ridges and valleys. The ridges are the dark area of the fingerprint and the valleys are the white area that exists between the ridges. Many classifications are given to patterns that can arise in the ridges and some examples are given in the figure to the right. These points are also known as the minutiae of the fingerprint. The most commonly used minutiae in current fingerprint recognition technologies are ridge endings and bifurcations because they can be easily detected by only looking at points that surround them.



Minutiae Matching

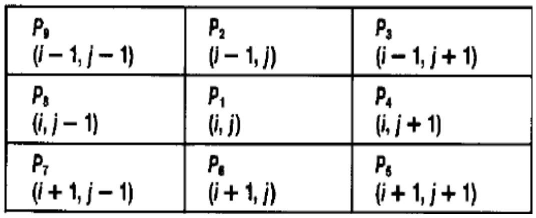
Most modern fingerprint matching technologies use minutiae matching. The idea being if you can find enough minutiae in one image that have corresponding minutiae in another image then the images are most likely from the same fingerprint. Minutiae are usually matched together by their distance relative to other minutiae around it. If multiple points in one image have similar distances between them then multiple points in another image then the points are said to match up. It is the idea of this paper to add the constraint that the regions and possibly edges between the minutiae should be the approximately the same as well.

The Process

Before minutiae matching can be done with the region information the fingerprint must be preprocessed. The preprocessing of the image is done in four stages. First the fingerprint is thinned. After a fingerprint is thinned it is easier to find minutiae such as bifurcations and endpoints simple. After thinning edge enhancement is used to fill in gaps in ridge detail. The gaps can be result of the thinning algorithm or the quality of the input fingerprint image. After the gaps have been filled in the region coloring begins. Sometimes spurious regions arise from the thinning process due to some conditions of the initial input, therefore after the regions are found small erroneous regions are removed from the picture

Thinning

Thinning was done using the Zhang-Suen algorithm as described in their paper title, *A Fast Parallel Algorithm for Thinning Digital Patterns*. A 3x3 window is move down throughout the image and calculations are carried out on each pixel to decide whether it needs to stay in the image or not. To the right is a description of the window and the classification given to the pixels that surround the center pixel. The algorithm runs two sub-iterations continuously until the image reaches a stable state.



P_9 $(i-1, j-1)$	P_2 $(i-1, j)$	P_3 $(i-1, j+1)$
P_8 $(i, j-1)$	P_1 (i, j)	P_4 $(i, j+1)$
P_7 $(i+1, j-1)$	P_6 $(i+1, j)$	P_5 $(i+1, j+1)$

Pseudo Code for Zhang –Suen Thinning:

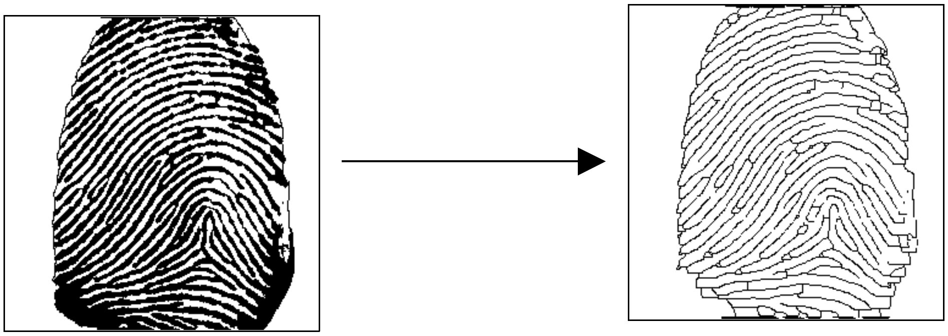
```
Let A(P) be the number of 01 patters in the order set $P_2 \ldots P_9$
Let B(P) be the number of non-zero neighbors of P

Do until image is stable (i.e. no changes made)
    Sub-iteration 1:
        Delete P from image if:
            a)  $2 \leq B(P) \leq 6$
```

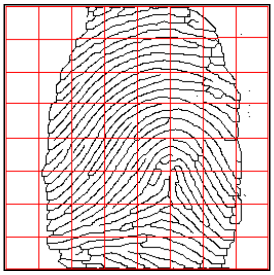
```
                b)  $A(P) = 1$
                c)  $P_2 * P_4 * P_6 = 1$
                d)  $P_4 * P_6 * P_8 = 1$
    Sub-iteration 2:
            Delete P from image if:
                a)  and b) from above
                c') $P_2 * P_4 * P_8 = 1$
                d') $P_2 * P_6 * P_8 = 1$
```

This processes effectively thins the image, however, it sometimes creates undesirably artifacts. In the example of a Zhang-Suen thinned fingerprint there are gaps between edges as well as regions of small area that need to be removed for proper regional analysis.

Example of Zhang-Suen Thinned fingerprint:



Scanning a Fingerprint



Many of the algorithms require a linear scan of the fingerprint image. Scanning is achieved by moving a fixed size window across the picture in a grid-like pattern. This can be seen in the image to the right. However, it is possible that areas of interest do not lie squarely in the one of the windows. To account for this the window is then shifted just vertically, just horizontally, and then vertically and horizontally by half the window size and the grid scan is completed again. Therefore, it takes four scans of the image to do the linear scan. This is not a problem because it is used for the preprocessing of a fingerprint image (which only occurs once) and is done in a linear manner.

Edge Enhancement

Regions are defined by the fingerprint edges that bound them. However, because of the nature of the fingerprint and current scanning technologies ridge detail can be missing from the scanned fingerprint. Furthermore, the thinning algorithm can also eliminate some of the edge detail. Most missing edges take the form of gaps in an edge and are usually easily identified by a human. The problem is getting a computer to recognize them. In the course of the project we developed our own edge enhancement technique, which caught some of the simple gaps, however, did not fill in the bigger or more complex ones. Therefore, I took part in the edge enhancement process by putting in some of the edges that were in the original picture but got dropped by the thinning process.

Pseudo Code for Edge Enhancement:

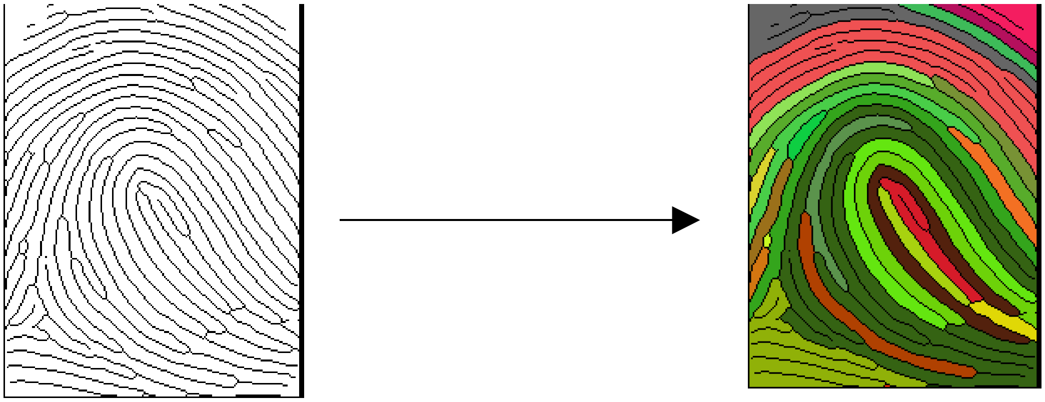
```
Scan Image as explained in previous section:
    For each window:
        Find all the endpoints in the window
    For each endpoint pair
        Look at the line that goes through the points
        If the line is "strong"
            Draw the line between the points
```

A line is said to be strong if many points on the line are already in or near points in the existing edge detail. If the line is strong then it is most likely that the ridge was supposed to span the gap and therefore can be drawn in. Note that because the image has been thinned it is easy to find endpoints in the image. Endpoints are merely pixels that only have one neighboring pixel (and are not at the edge of the window).

Region Coloring

After the edges have been properly added to the image as necessary the region coloring can begin. The region coloring is similar to taking the image into Microsoft Paint and using the color fill tool on all the valleys of the fingerprint, however, the code does it in a little more efficient way. The image is scanned and colors (represented by numbers) are associated with white space in the image. While the image is being colored in each pixel looks at the pixels around it to determine what color it should be and color equivalences are set up. After the image is done being scanned, it is scanned again this time replacing all the colors in the equivalence set with just one of the colors from them. Therefore, each pixel in a region has the same color number associated with it.

Example of Region coloring:



Spurious Region Removal

As an artifact of the thinning process small regions are created where there should be no region. Because the matching process relies on the region preprocessing to be accurate these regions must be removed

Pseudo Code for Region Removal

```
For each window in scan:
    For each region encased in the window that is sufficiently small
        Remove the border between the erroneous region and the region it
        borders with least
```

This will make it so that regions that are small are spilled into a surrounding region that it most likely was supposed to belong to.

Testing

After the four steps of preprocessing are completed the image can be used for the matching process. However, first we had to show that the region data was a good indication of matching minutiae. That is we need to show that the amount of regions between two minutiae is consistent across many different scans of the same fingerprint. To test our hypothesis 8 different scans of the same finger were used. Two separate minutiae pairs were used and the corresponding minutiae pairs were found on each of the 8 scans. The amount of edges and the amount of regions between the chosen minutiae points were calculated and the results were compared.

Input Fingerprints:

Image 1

Image 2

Image 3

Image 4

Image 5

Image 6

Image 7

Image 8

The preprocessing algorithms were run on each of the 8 different scans of the same finger to obtain the output fingerprints:

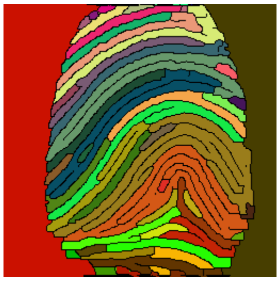


Image 1

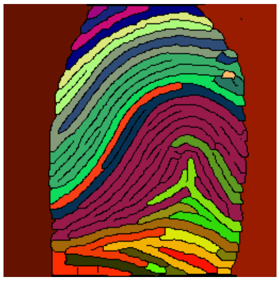


Image 2

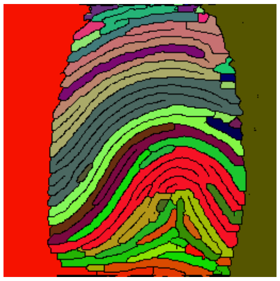


Image 3

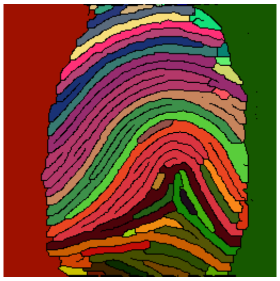


Image 4

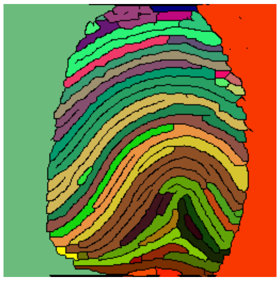


Image 5

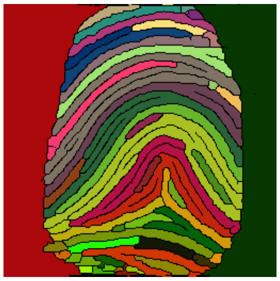


Image 6

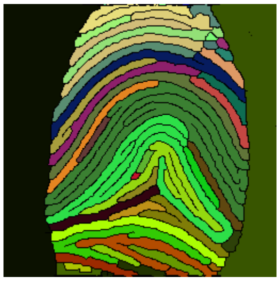


Image 7

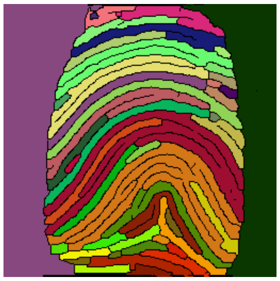
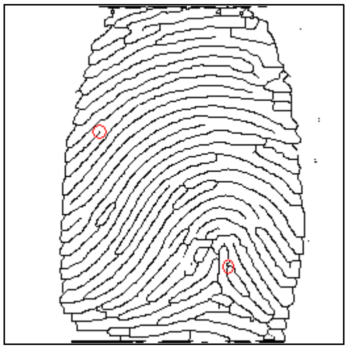
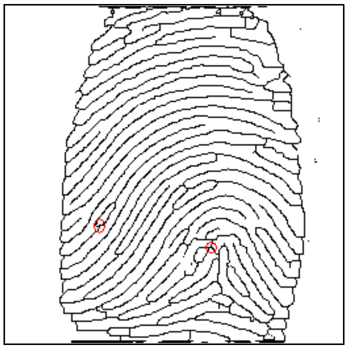


Image 8

The calculations were then drawn based on the two chosen minutiae pairs:



Pair 1



Pair 2

Results

Minutiae Pair 1

Image Number	Regions Between Minutiae	Edges Between Minutiae
1	13	20
2	9	16
3	11	19
4	11	17
5	12	19
6	10	16
7	11	15
8	13	8

Statistics:

- Max Region Difference:	4
- Average Region Difference:	1.64
- Max Edge Difference:	5
- Average Edge Difference:	2.21

Minutiae Pair 2

Image Number	Regions Between Minutiae	Edges Between Minutiae
1	7	11
2	5	13
3	6	11
4	7	12
5	7	11
6	6	10
7	4	8
8	7	12

Statistics:

- Max Region Difference:	3
- Average Region Difference:	1.25
- Max Edge Difference:	5
- Average Edge Difference:	1.7

Note: Average differences were found by taking the average distances when matching all the images into pairs.

Analysis

For both the sets of minutiae the average number of region difference and edge difference is between 1 and 3. This is a reasonable number because of errors that arrive from the initial input and the preprocessing algorithm. Basically, the matching algorithm that uses the regional data would have to take into account that the region count between two minutiae can be off by a certain threshold. A reasonable threshold to use would be 2 or 3 regions as this would work for most of the pairs of images.

Future work:

Future work on this project would include the creating of a matching algorithm that uses the regional data created in this preprocessing system. The matching algorithm would have to be both efficient in theory and the code that implements it because the goal would be to make it run faster and more accurate then pre-existing software.

Also, more work can be done on the edge enhancing algorithm as it does not properly detect missing edges that are supposed to be curved. A way this can be accomplished is to take the edge orientation (the way an edge is curving) into consideration.

References

Bhanu Bir, Tan Xuejun, Computational Algorithms for Fingerprint Recognition. USA: Kluwer Academic Publishers, 2004

Das, K. *Design and Implementation of an Efficient Thinning Algorithm*. Bachelor of Technology thesis, Department of Computer Science and Engineering, Indian Institute of Technology, Kanpur. April 2000.

Jain Anil K et al. Biometrics: Personal Identification in Networked Society. Springer, 1999

Lei, Guo et al. "Characteristic Preserving Binarization for Fingerprint Image". Image and Graphics. August 2007, Pages 401- 408

O'Gorman L., "Overview of fingerprint verification technologies," Elsevier Information Security Technical Report, Vol. 3, No. 1, 1998.

Zhang T. Y, Suen C. Y, "A Fast Parallel Algorithm for Thinning Digital Patterns". Communications of the ACM. March 1984, Pages 236 - 239

Acknowledgements

I would like to thank Professor Rafail Ostrovsky for his input and advice on the project as well as contributing to the design of the algorithms used.

I would also like to thank Professor Amit Sahai for the opportunity to take part in the research program.