Fingerprint Minutiae Extraction and Compression Using LZW Algorithm

Balpreet Singh[1] Ishpreet Singh[2]
[1]M.Tech Student [2]Assistant professor
[1,2]Department of Computer Science and Engineering
[1,2]BBSBEC Fatehgarh Sahib

Abstract— For security and surveillance automated personal identification is major issue. We can see a lot varieties of biometric systems like face detection, fingerprint recognition, iris recognition, voice recognition, palm recognition etc. In our project we will only go for fingerprint recognition. Never two peoples have exactly same fingerprints even twins, they are totally unique. The sensors capture the finger prints of humans and convert them into images and a minutiae extraction algorithm extracts the location of minutiae points called termination and bifurcation. A database system stores these patterns and minutiae points of fingerprint. A large storage space required to store bifurcation and termination points for the fingerprint database. LZW compression algorithm has been used to reduce the size of data. With LZW applied on these extracted minutiae points, these minutiae points get encoded which add more security feature..

Key words: Fingerprint, Bifurcation and Termination, Minutia, LZW Algorithm.

I. INTRODUCTION

Fingerprint used for identification are obtained from sensors or fingerprint machine. Fingerprints are more secure than the password and never two people have been found with the exact same fingerprint pattern. A Key's and id-cards are mostly used parameter for personal identification and also are unbearable attacks. Fingerprint identification, it is commonly employed in cyber-crime to aid criminal investigations. Ridges and valleys are consisting on the skin of human fingertips and they mixed together to the distinctive patterns. Fingertips on human being is very unique pattern recognition problem so manual finger print matching is not only time taking but experts also required long time for education and training. Fingerprints have remarkable permanency and uniqueness throughout the time. Example such as mobiles and computers, now in these days the sensor system is mostly implemented as compare to passwords because it is most secure than password.

A. Fingerprint

Every day we touch a lot of things like a mobile, a cup of coffee and every day we leave unique signature like fingerprints behind them. Fingerprints of human being are unique and it is very difficult to alter. Many crimes have being solved with the help of fingerprint analysis and it is valuable tool for law enforcement. The skin on the fingertips consists of ridge and valley which are unique to each individual and which do not change over time. So fingerprints have been used for the forensic application and identification for a long time. There are many types of fingerprints readers in market but the basic idea behind them is abstract ridge and valley from the fingertips.

It can be distinguished by Minutia, which is some abnormal points on the ridges. Minutia is divided in to two parts such as: termination and bifurcation. Termination is also called ending and bifurcation is also called branch. Again minutia consists of ridges and furrows.

Fig. 1.1: Finger print image acquired by a Sensor

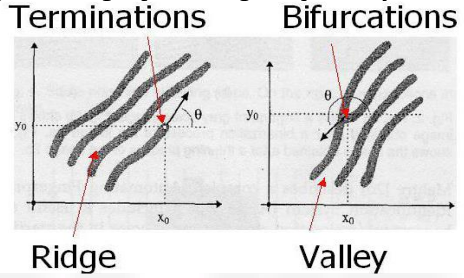

Fig. 1.2: Specific Termination & Bifurcation

B. Fingerprint Recognition:

The fingerprint recognition problem can be grouped into two sub-domains such as:-

- Fingerprint verification
- Fingerprint identification

Fingerprint verification is the method where we compare a claimant fingerprint with an enrollee fingerprint, where our aim is to match both the fingerprints. This method is mainly used to verify a person’s authenticity. For verification a person needs to his or her fingerprint in to the fingerprint verification system. Then it is representation is saved in some compress format with the person’s identity and his or her name. Then it is applied to the fingerprint verification system so that the person’s identity can be easily verified. Fingerprint verification is also called, one-to-one matching.

Fingerprint identification is mainly used to specify any person’s identity by his fingerprint. Identification has been used for criminal fingerprint matching. Here the system matches the fingerprint of unknown ownership against the other fingerprints present in the database to associate a crime with identity. This process is also called, one to many matching. Identification is traditionally used for solve crime and catch thieves.

II. MINUTIA BASED IMPLEMENTATION:

Minutiae based fingerprint recognition is most researched and modern technique. A critical step in automatic fingerprint matching is to automatically and reliably extract minutiae from the input fingerprint images. However, the performance of a minutiae extraction algorithm relies heavily on the quality of the input fingerprint images. Image enhancement algorithm needs to keep the original ridge flow pattern without altering the singularity, join broken ridges, clean artifacts between pseudo-parallel ridges, and not introduce false information.

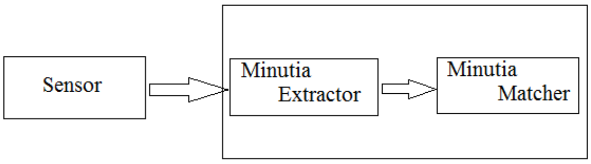

Fig. 2.1: Fingerprint Acquiring device, Minutia extractor and minutia matcher

A. Fingerprints Matching Technique

Minutiae-based Matching: This is the most widely used technique. This method uses the local features of fingerprint. Each Minutia point is described by a number of attributes, including its location in the fingerprint image, (e.g., ridge termination or ridge bifurcation). Minutiae are extracted from the two fingerprints and stored as sets of points in the two dimensional plane. Minutiae based matching essentially consists of finding the alignment between the stored image and the input with respect to the minutiae points. Minutia based fingerprint representation and matching are widely used by both machine and human experts because of several advantages compared to other fingerprint representations. Such as, its configuration is highly distinctive and minutiae based systems are more accurate than correlation based systems and the template size of minutia based fingerprint representation is small. Minutia based fingerprint representation also has an advantage in helping privacy issues, since one cannot reconstruct the original image from using only minutiae information. But, reliably extracting minutia from poor quality fingerprints is very difficult and minutiae extraction is also very time consuming. Because of the advantages of minutia based fingerprint matching and as it is the backbone of the current available fingerprint recognition products, in this paper 'Minutia based matching' was applied.

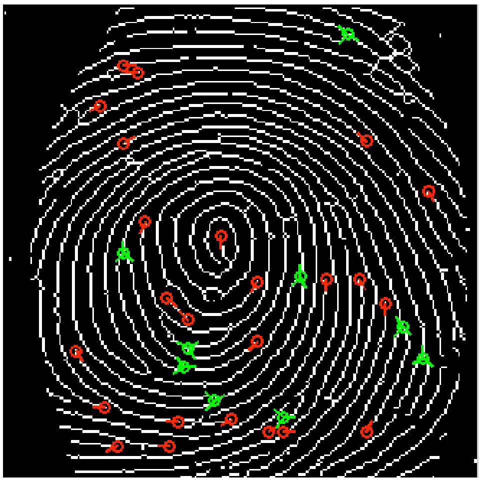

Fig. 2.2: Extracted Minutiae Points

B. Minutiae Extraction

The next step is minutiae extraction which consists of series of action. Among all the fingerprint features, minutia point features with corresponding orientation maps are unique enough for making distinctions. The minutiae feature representation reduces the complex fingerprint recognition problem to a point pattern matching problem. An accurate extraction and representation of the fingerprint feature is very important in automatic fingerprint recognition systems. Minutiae detection algorithm needs to locate efficiently and accurately the minutiae points. There are various minutiae extraction algorithm available, they can be categorized into four groups. The first type of groups extracts minutiae points directly from the gray-scale image. A second type of methods extracts minutiae from binary image. Third type of methods extracts minutiae using machine learning methods. The last type of methods extracts minutiae from binary skeletons. In paper we have concentrated on Skeletonization-based Minutiae Extraction. This method has four subparts as described below.

- Ridge Thinning
- Minutiae Marking
- False Minutiae marking
- Minutiae Point Representation

Fig. 2.3: Ridge Thinning

C. Minutiae Marking:

This is the most important step in the fingerprint recognition because at this stage features are extracted from the fingerprints. Minutiae marking follow the ridge thinning process. Extracting the minutiae from the one-pixel wide ridge map is a vital process. Minutiae detection in a fingerprint image is implemented by scanning the thinned fingerprint image and method of Crossing Number (CN) is used. The concept of the process is as follow.

P4	P3	P2
P5	P	P1
P6	P7	P8

Fig. 2.4: 3x 3 Windows

The CN value is computed, which is defined as half the sum of the differences between pairs of adjacent pixels in the eight Neighborhoods. Using the properties of the CN as shown in table, the ridge pixel can then be classified as a ridge ending, Bifurcation or non-minutiae point. For example, a ridge pixel with a CN of one corresponds to a ridge ending, and a CN of Three corresponds to a bifurcation. The CN is given by Eq. 2.1 The minutiae points detected using CN number are represented using techniques described in following section and stored in file. All these minutiae points are called candidate minutiae points, in the following section season will become clear why we call them candidate minutiae point.

$$CN=\frac{1}{2}\sum_{i=0}^{8} Pi - Pi + 1 \qquad \text{Eq. 2.1}$$

1) Rule 1:

If central pixel is 1 and has exactly 3 one- value neighbors, then central pixel is ridge bifurcation.

0	1	0
0	1	0
1	0	1

Fig. 2.5: Ridge Bifurcation Structure

2) Rule 2:

If central pixel is 1 and has exactly 1 one-value neighbors, then central pixel is ridge Termination.

0	0	0
0	1	0
0	0	1

Fig. 2.6: Ridge Termination Structure

The minutiae points detected using CN number are represented using techniques described in following section and stored in File. All these minutiae points are called candidate minutiae points, in the following section season will become clear why we call them candidate minutiae point.

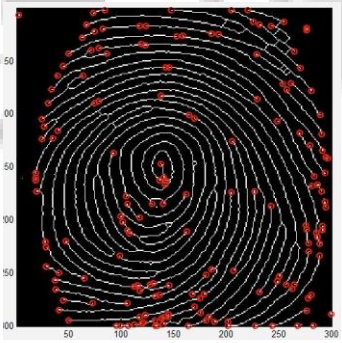

Fig. 2.7: Termination Points

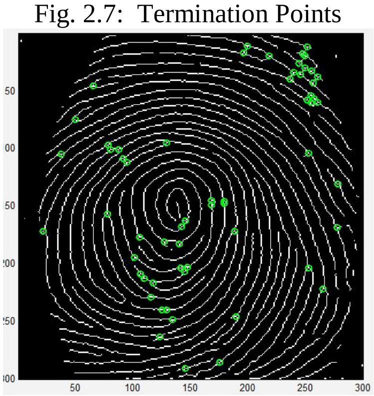

Fig. 2.8: Bifurcation Points

X	Y	Angle
230	86	2.36
206	126	-0.79
138	147	-1.05
137	163	2.09
227	173	-1.57
205	174	-2.09
162	176	-2.36
106	185	-1.57
243	187	-2.09
102	202	-1.05
163	211	-2.09
207	248	0.79
181	252	-2.36
131	260	0.00
146	262	-2.62
232	268	1.05
168	270	0.52

Table1: Data Extract from Fig. 2.7

X	Y	Angle
130	95	2.36
88	101	2.62
78	157	-2.36
189	172	-2.62
253	204	-2.36
118	217	2.62
116	229	-2.62
135	249	2.62

Table 2: Data extract from Fig. 2.8

III. MOTIVATION FOR THE WORK

In a large database of images e.g. UID database, number of images can be up to millions. Minutiae points extracted from these images for identification and verification also require a lot of space and also are vulnerable to security attack. So motivation behind this work is to compress the minutiae points extracted from the images and we use LZW Algorithm to compress the data.

A. Advantages of Proposed Work

- It works well for images of any bit depth, including images deeper than 8 bits per sample.
- It does not depend on small variations between pixels.
- It should not take more than 5 seconds to decompress a 100K byte grayscale image.
- The level of implementation complexity is reasonable.
- LZW is fully reversible.
- LZW decompression speeds about 50K bytes per second.

IV. DIFFERENT TYPES OF DATA COMPRESSION:

Data compression is the process of encoding information using fewer bits or other information-bearing units. Compression is possible where the input data have statistical redundancy (e.g., in text files) or when relatively minor changes leading to smaller representation do not affect the quality/fidelity of the input (e.g., in pictures, video, or audio files). Compression is important because it helps reduce the consumption of expensive resources, such as disk space or connection bandwidth. However, compression requires increased information processing power, which can also be expensive.

There are basically two types of compression techniques. The one compression technique is lossless and the technique of compression is lossy technique.

A Lossless compression technique performs full reconstruction of the original data without incurring any distortion in the process. The Lossless Data Compression technique recommended preserves the source data accuracy by removing redundancy from source data. The quantity of redundancy removed from the source data is variable and highly dependent on the source data statistics. It may be necessary to rearrange the data into appropriate sequence before applying the data compression algorithm. The purpose of rearranging data is to improve the compression ratio.

Lossy compression is compression in which some of the information from the original message sequence is lost. This means the original sequences cannot be regenerated from the compressed sequence. When quality does start degrading in a noticeable way, it is important to make sure it degrades in a way that is least objectionable to the viewer (e.g., dropping random pixels is probably more objectionable than dropping some color information).

A. *LZW Technique*

- The Lempel-Ziv-Welch (LZW) compression algorithm is an example of dictionary based methods, in which longer fragments of the input text are replaced by much shorter references to code words stored in the special set called dictionary.
- LZW is an implementation of a lossless data compression algorithm developed by Abraham Lempel and Jacob Ziv.
- The key insight of the method is automatically building a dictionary of previously seen strings in the text being compressed.
- The dictionary starts off with 256 entries, single byte string.
- These initially are 9 bits each, and as the dictionary grows, can increase to up to 16 bits.
- A special symbol is reserved for "flush the dictionary" which takes the dictionary back to the original 256 entries, and 9 bit indices. This is useful if compressing a text which has variable characteristics, since a dictionary of early material is not of much use later in the text.

B. *Working of LZW Algorithm*

LZW compression replaces strings of characters with single codes. It does not do any analysis of the incoming+ text. Instead, it just adds every new string of characters it sees to a table of strings. Compression occurs when a single code is output instead of a string of characters. The code that LZW algorithm outputs can be of any arbitrary length but it must have more bits in it than a single character. The first 256 codes are by default assigned to standard character set. The remaining codes are assigned to strings as the algorithm proceeds. This means codes 0-255 refer to individual bytes whereas codes 256-4095 which is the maximum limit of the dictionary refers to sub strings.

1) *Encoding using LZW*

- Table of Character and Strings
- i= first input String
- WHILE not end of input stream
- j=next input character
- IF i+j is in the string table
- i=i+j
- ELSE
- output the code for i
- add i+j to the string table
- i=j
- END WHILE
- output code for i

2) *Decoding using LZW*

Initialize table with single character strings

- OLD = first input code
- output translation of OLD
- WHILE not end of input stream
- NEW = next input code
- IF NEW is not in the string table
- i = translation of OLD
- i = i+j
- ELSE
- i = translation of NEW
- output S
- j = first character of S
- OLD + j to the string table
- OLD = NEW
- END WHILE

C. *Comparison of LZW with Huffman Coding*

- Huffman coding is statistical based coding whereas LZW is dictionary based coding.
- Huffman coding is a fixed to variable length code. LZW on the other hand is a variable to fixed length code.
- The design of the Huffman code is optimal assuming that the source statistics are known a prior. The LZW code is not designed for any particular source but for a large class of sources.
- Huffman coding technique requires a first pass to analyze the characteristics of the source. Whereas LZW coding is a free-parse method. That is the words of the source alphabet are defined dynamically as the encoding is performed.

V. THE PROPOSED SYSTEM

The fingerprints image for any person can be utilized to generate unique code. The fingerprint image passes through a sequence of operations that includes scan image fingerprint, create image of finger print then get termination and bifurcation from that image after that create LZW algorithm on that data.

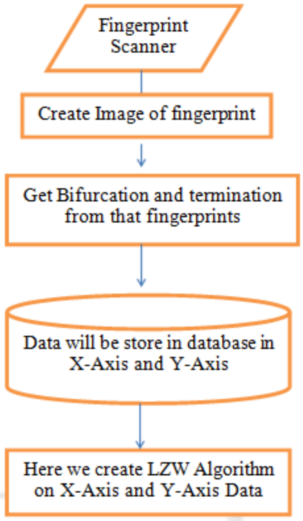

Fig. 5.1: Flow diagram for proposed work

So we can say that in this proposed work we are able to achieve a compression rate up to 50%. Nest is graphical representation of results obtained by proposed work.

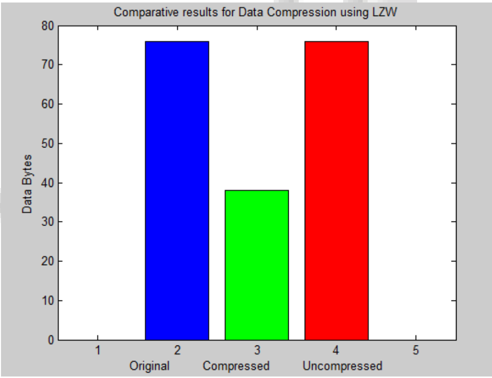

Fig. 5.2: Flow Chart Extract from Table 1

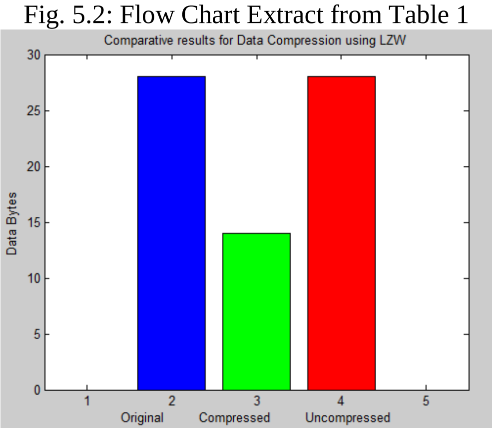

Fig. 5.3: Flow Chart Extract from Table 2

VI. CONCLUSION

The efficiency of any automatic fingerprint system strongly relies on the precision obtained in the minutia extraction. Minutiae are generally used for security issues and classification purposes. But to degrade the classification of minutia there are many factors on which it depends. Among them, storing the feature points is the one with most influence. Good storage in database of fingerprint images features needs best and enhancement algorithm of data encoding for accurate identification algorithm. In this paper, compression and encoding using LZW technique is used for compression of features extracted using Morphological operations, minutia marking by specially considering branch counting, branch into terminations.

Adding LZW Algorithm to reduce the storing capacity of fingerprint images features into the database is helpful in making system more stable and efficient. There is a scope of further improvement in terms of efficiency and accuracy, which can be achieved by improving the compression and encoding algorithm. So that the database can save more data for long time which could improve the future stages.

REFERENCES

[1] Ishpreet Singh Virk, Raman Maini," Finger Print Image Enhancement And Minutiae Matching In Finger Print Verification" Journal Of Computing Technologies, Vol. 2, pp.1-14 ISSN 2278-3814 2012.
[2] Vipul Awasthi, Vanchha Awasthi and Krishna Kumar Tiwari, "Finger Orint Analysis Using Termination and Bifurcation Minutiae", pp. 2250-2459, Volume 2, Issue 2, February 2012.
[3] Ankit Sharma, Vinod Kumar and Surabhi Jain, " Survey on Minutiae Coordinates of X-Axis and Y-Axis in Fingerprint Captured Image using Optical Fingerprint Scanner", pp.1715-1718, Vol. 3, Issue 2, March -April 2013.
[4] YonghuiWu, Stefano Lonardi and Wojciech Szpankowski, "Error-Resilient LZW Data Compression", IEEE Proceedings of the Data Compression Conference, 2006.
[5] Robert Hastings, "Ridge Enhancement in Fingerprint Images Using Oriented Diffusion", The University of Western Australia, Australia. pp. 245-252, IEEE 2007.
[6] Dr. Bashar M. Nema and Nuha Sami Muhsin, "Generate a Secured Multi-Purpose Code from Fingerprint Images", Al-Mansour Journal Issue, pp. 23-34, 2011.
[7] Manvjeet Kaur, Mukhwinder Singh, Akshay Girdhar, and Parvinder S. Sandhu, "Fingerprint Verification System using Minutiae Extraction Technique pp.432-437 Vol. 2 2008.
[8] Chitresh Saraswat, and Amit Kumar, "An Efficient Automatic Attendance System using Fingerprint Verification Technique" pp. 264-269, Vol. 02, 2010,
[9] G.Nagendra V.Sridhar M.Prame Elamma and M.Zubair, "Fingerprint Image Enhancement Using Filtering Techniques" pp 61-78 Vol. 1 No.01 September 2012.
[10] Roli Bansal, Priti Sehgal and Punam Bedi, "Minutiae Extraction from Fingerprint Images", pp.74-85 Vol. 8, Issue 5, No 3, September 2011

[11] Bashar Ne'ma and Hamza Ali, "Multi Purpose Code Generation Using Fingerprint Images", Software Engineering Department, Isra Private University, Jordan pp. 418-423, OCTuber 2009.
[12] Mehnaz Tabassum, "A New Method for Biometric Based Recognition", pp.1-7, Volume 3, Issue 9, September 2013.
[13] Agus Dwi Suarjaya, "A New Algorithm for Data Compression Optimization", Information Technology Department Udayana University Bali, Indonesia pp. 14-17 Vol. 3, No.8, 2012.
[14] M. Sepasian, W. Balachandran and C. Mares, "Image Enhancement for Fingerprint Minutiae-Based Algorithms Using CLAHE, Standard Deviation Analysis and Sliding Neighborhood" October 22 - 24, 2008.
[15] Kulwinder Singh, Kiranbir Kaur, Ashok Sardana, "Fingerprint Feature Extraction "Vol. 2, Issue 3, September 2011.
[16] Kodituwakku, U.S. Amarasinghe, "Comparison of Lossless Data Compression Algorithms for text data ", Vol 1 No 4 PP. 416-425 PP. 22 – 24.
[17] L. Ravi Kumar et S. Sai Kumar, J. Rajendra Prasad, B. V. Subba Rao, P. Ravi Prakash , " Fingerprint Minutia Match Using Bifurcation Technique " Vol 2(4), PP. 478-486.
[18] Saud NAQVI, Rameez NAQVI, Raja Ali RIAZ, Faisal SIDDIQUI, "Optimized RTL design and implementation of LZW algorithm for high bandwidth applications" ISSN 0033-2097, R. 87 NR PP. 279-285 April 2011.
[19] Amit Makkar, Gurjit Singh, Rajneesh Narula, " Improving LZW Compression" PP. 411-414 IJCST Vol. 3, Issue 1, Jan. - March 2012.
[20] Dheemanth H N, "LZW Data Compression ", Volume-03, Issue-02, pp-22-26 American Journal of Engineering Research (AJER) 2014.
[21] Simrandeep kaur, " HDL Implementation of Data Compression: LZW algorithm " pp 115-120 Volum`e 2, Issue 2, March 2012.
[22] Shyni, Manoj Kumar, " Lossless LZW Data Compression Algorithm on CUDA", Volume 13, Issue 1 (Jul. - Aug. 2013), PP 122-127.