ISSN Print: 2394-7500
ISSN Online: 2394-5869
Impact Factor: 5.2
IJAR 2015; 1(7): 32-38
www.allresearchjournal.com
Received: 15-04-2015
Accepted: 14-05-2015

Gowtham M
PG Scholar, SNS College of Technology, Coimbatore-641 035, Tamilnadu, India.

Arulkarthick V J
Assistant Professor, SNS College of Technology, Coimbatore-641 035, Tamilnadu, India.

Gokilavani Chinnasamy
PG Scholar, SNS College of Technology, Coimbatore-641 035, Tamilnadu, India.

Correspondence:
Gowtham M
PG Scholar, SNS College of Technology, Coimbatore-641 035, Tamilnadu, India.

A Novel Fingerprint Image Compression Technique Using Run-Length and Entropy Coding

Gowtham M, Arulkarthick V J, Gokilavani Chinnasamy

Abstract
Image compression is currently a prominent topic for both military and commercial researchers. Due to rapid growth of digital media and the subsequent need for reduced storage and to transmit the image in an effective manner Image compression is needed. Image compression attempts to reduce the number of bits required to digitally represent an image while maintaining its perceived visual quality. Biometrics has physiological characteristics means it includes of latent fingerprint recognition and verification and behavioral characteristics means it includes voice and signature. A new fingerprint compression algorithm based on Run-Length Encoding (RLE) with Patch Separation representation is introduced. RLE is based on the idea to replace a long sequence of the same symbol by a shorter sequence and is a good introduction into the image compression field for newcomers. The RLE Finger print compression algorithm first construct a Independent Component Analysis (ICA) Patch Extraction for predefined fingerprint image patches to overcome the dictionary creation. RLE is suited for image compressing any type of Latent fingerprint regardless of its information content, but the content of the data will affect the compression ratio achieved by RLE.

Keywords: Biometrics, Fingerprint, Independent Component Analysis, Latent, Patches, Run-Length Encoding.

1. Introduction

Recognition of persons by means of biometric characteristics is an important technology in the society, because biometric identifiers can't be shared and they intrinsically represent the individual's bodily identity. Among many biometric recognition technologies, fingerprint recognition is very popular for personal identification due to the uniqueness, universality, collectability and invariance [1]. Large volumes of fingerprint are collected and stored every day in a wide range of applications, including forensics and access control. In 1995, the size of the FBI fingerprint card archive contained over 200 million items and archive size was increasing at the rate of 30 000 to 50 000 new cards per day [1]. Large volume of data consumes the amount of memory. Fingerprint image compression is a key technique to solve the problem.

The central problem of this thesis comes from the field of biometric data processing. The term biometrics (Greek: Bio life, Metron measure) collects many procedures of measuring and analyzing humans based on one or more physical and behavioral traits. Of the many fields of which biometrics is comprised in this diploma thesis the procedures of biometric recognition processing are of interest. The origins of biometrics are located in forensic sciences, dactyloscopy and cryptography. In the past decades, with the progression of information technology, the interest and distribution of biometric applications ever increased, since cataloguing and analyzing huge amounts of data became only possible with recent technology.

Biometric recognition bases upon several traits and characteristics of the human body and behavior, named biometric identifiers. According to [MalJai05] these identifiers are categorized into physiological (fingerprints, iris, retina, vein structure, DNA, teeth alignment, hand, palm and face geometry) and behavioral (written signature, gait, voice, typing rhythm). It describes several properties that such a trait must surface, in order to be used as a biometric identifier:

- Universality: The characteristic is found on every individual to be verified.
- Uniqueness: The characteristic is sufficiently different across the individuals.
- Permanence: The characteristic remains constant over time, and depends not on age or measure datum.
- Measurability: The characteristic is acquirable with a suited device, without incommoding the providing person. Digitalizing and further processing must be possible.
- Performance: Accuracy and resource consumption for acquiring and processing should have acceptable time, computation and hardware effort.
- Acceptability: Persons should be willing to present their biometric trait to the system.
- Circumvention: The characteristic should not be easily imitated, reproduced or acquired.

2. Fingerprint Compression Model

The dictionary based coder approach named above is then pursued in the following way: Having in mind the construction of a minimal dictionary or codebook for a compact representation of minutiae information, the naturally contained structures and redundancies in minutiae data led to the development of a graph based encoding model as a basis for the compression mechanism.

Out of this graph model the compression algorithm extracts the dictionary in form of so called template vectors and afterward expresses the minutiae from the input data by a reference to the appropriate codebook entry. The codebook or dictionary is schematized in figure 1.1. This emphasize that approach follows this general idea of using a dictionary or codebook but its determination as well as its usage is very different to the existing methods.

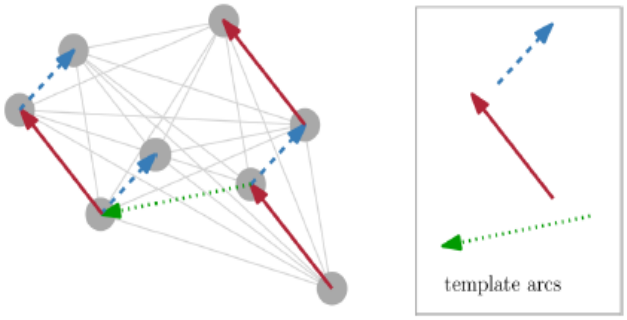

Fig 2.1: Encoding Of Points Via A Directed Spanning Tree Using A Codebook Of Template Arcs, Correction Vectors Are Neglected.

Law enforcement agencies have started using fingerprint recognition technology to identify suspects since the early 20th century [2]. Nowadays, automated fingerprint identification system (AFIS) has become an indispensable tool for law enforcement agencies.

There are essentially three types of fingerprints in law enforcement applications (see Fig. 2): (i) rolled, which is obtained by rolling the finger "nail-to-nail" either on a paper (in this case ink is first applied to the finger surface) or the platen of a scanner; (ii) plain, which is obtained by placing the finger flat on a paper or the platen of a scanner without rolling; and (iii) latents, which are lifted from surfaces of objects that are inadvertently touched or handled by a person typically at crime scenes. Lifting of latents may involve a complicated process, and it can range from simply photographing the print to more complex dusting or chemical processing.

Rolled prints contain the largest amount of information about the ridge structure on a fingerprint since they capture the largest finger surface area; latents usually contain the least amount of information for matching or identification because of their size and inherent noise. Compared to rolled or plain fingerprints, latents are smudgy and blurred, capture only a small finger area, and have large nonlinear distortion due to pressure variations. Due to their poor quality and small area, latents have a significantly smaller number of minutiae compared to rolled or plain prints (the average number of minutiae in NIST Special Database 27 (NIST SD27) [3] images is 21 for latents versus 106 for their mated rolled prints). These characteristics make the latent fingerprint matching problem very challenging.

Fingerprint examiners who perform manual latent fingerprint identification follow a procedure referred to as ACE-V (analysis, comparison, evaluation and verification) [4]. Because the ACE-V procedure is quite tedious and time consuming for latent examiners, latents are usually matched against full prints of a small number of suspects identified by other means, such as eye witness description or M.O. (mode of operation).With the availability of AFIS, fingerprint examiners are able to match latents against a large fingerprint database using a semiautomatic procedure that consists of following stages: (i) manually mark the features (minutiae and singular points) in the latent, (ii) launch an AFIS search, and (iii) visually verify the top- (is typically 50) candidate fingerprints returned by AFIS. The accuracy and speed of this latent matching procedure is still not satisfactory. It certainly does not meet the "lights-out mode" of operation desired by the FBI and included in the Next Generation Identification [5]. In [6-18], various image compression standards, techniques and algorithms have been discussed.

(a) (b) (c)

Fig 2.2: Three Types of Fingerprint Impressions (a) Rolled; (b) Plain; (c) Latent.

For fingerprint matching, there are two major problems which need to be solved. The first is to align the two fingerprints to be compared and the second is to compute a match score between the two fingerprints. Alignment between a latent and a rolled print is a challenging problem because latents often contain a small number of minutiae and undergo large skin distortion. To deal with these two problems, we propose the descriptor-based Hough transform (DBHT), which is a combination of the generalized Hough transform and a local minutiae descriptor, called Minutia Cylinder Code (MCC). The MCC descriptor improves the distinctiveness of minutiae while the Hough transform

method can accumulate evidence as well as improve the robustness against distortion. Match score computation between a latent and a rolled print is also challenging because the number of mated minutiae is usually small. To address this issue, we further consider orientation field as a factor in computing match score. Since we only have manually marked minutiae for latent's, a reconstruction algorithm is used to obtain orientation field from minutiae.

Automatic fingerprint recognition has become a widely used technology in both forensic and biometrics applications. Despite a history of a thousand years during which fingerprints have been used as individual's proof of identity and decades of research on automated systems, reliable fully automatic fingerprint recognition is still an unsolved challenging research problem. Moreover, most of the research thus far, assumes that the two fingerprint templates being matched are approximately of the same size and cover large areas of the finger tip. However, this assumption is no longer valid. The miniaturization of fingerprint sensors has led to small sensing areas and can only capture partial fingerprints. Partial fingerprints are also common in forensic applications.

Fingerprint matching based on minutia features is a well researched problem. During the last four decades, various algorithms have been proposed to match two minutia templates of fingerprints. Most of these algorithms assume that the two templates are approximately of the same size. This hypothesis is no longer valid. Miniaturization of fingerprint sensors has led to small sensing areas usually varying from 1"x1" to 0.42"x0.42". However, fingerprint scanners with a sensing area smaller than 0.5"x0.7", which is considered to be the average fingerprint size, can only capture partial fingerprints.

Matching small (partial) fingerprints to full pre-enrolled images in the database has several problems: (i) the number of minutia points available in such prints is few, thus reducing its discriminating power; (ii) loss of singular points (core and delta) is likely and therefore, a robust algorithm independent of these singularities is required; and (iii) uncontrolled impression environments result in unspecified orientations of partial fingerprints, and distortions like elasticity and humidity are introduced due to characteristics of the human skin.

A minutiae based fingerprint matching system usually returns the number of matched minutiae on both query and reference fingerprints and uses it to generate similarity scores. Generally, more matched minutiae yield higher similarity scores. That is when the number of minutiae on both fingerprints is large we can confidently distinguish the genuine and imposter fingerprint using the number of matched minutiae. According to forensic guidelines, when two fingerprints have a minimum of 12 matched minutiae they are considered to have come from the same finger. However, it is not reasonable to use an absolute number of matched minutiae alone in case of partial fingerprints. We must also consider the overlapped areas on both prints and the total distance between all the matched minutiae to obtain a similarity score.

Algorithms for fingerprint matching are used in human identification systems with applications in biometrics and forensics. The structures most widely used by fingerprint matching algorithms are minutiae–which are representations of ridge bifurcations and ridge endings. Minutiae-based algorithms with varying accuracy and efficiency are described in the literature on automatic fingerprint identification systems (AFIS).

In several scenarios only a partial fingerprint image is available as input, e.g., compact silicon chip-based sensors that capture only part of the fingerprint, processing latent fingerprints from crime scenes, etc. The task in such cases is to match an incomplete input fingerprint against pre-enrolled full fingerprints. While some minutiae matchers are highly accurate in full fingerprint matching, their error rates dramatically increase with decreasing number of minutiae.

The use of ridge similarity for fingerprint alignment has been previously described. While it leads to efficiency it is also vulnerable to non-linear deformation. When the threshold for ridge similarity is loosened to tolerate non-linear deformation, alignment errors will simultaneously increase. Even if the original scheme is improved by considering multiple ridge pairs for alignment, it is still worse than some matchers that use only minutiae [1]. In some minutiae matchers with high accuracy, local similarity models cover small regions. However, entire ridges cover relatively larger regions. Similarity defined on a larger region is apparently more sensitive to non-linear deformation– which prevents the ridge similarity model from better performance. An effective algorithm for using ridges, based on utilizing representative ridge points (RRPs), is the focus. A consideration was that an RRP have the same representation as minutiae so that existing minutiae matchers could be utilized with simple modifications.

In a live-scan AFRS, a user puts his/her finger against the prism and the contact fingerprint region will be captured in the resulting image. A small contact region between the finger and the prism will lead to a small partial fingerprint image. On such small fingerprint region, there could be very limited minutiae available for recognition. A natural way to solve the partial fingerprint recognition problem is to make full use of other fine fingerprint features abundant on the small fingerprint fragments. Sweat pores are such kind of features and high resolution fingerprint imaging makes it possible to reliably extract the sweat pores on fingerprints.

Most existing high resolution fingerprint recognition methods use full-size fingerprint images which capture large fingerprint areas. However, to capture the full fingerprints, high resolution fingerprint images should have much bigger sizes than conventional low resolution fingerprint images. As a result, much more computational resources are required to process the images. Considering the increasing demand of AFRS on mobile devices and other small portable devices, small fingerprint scanners and limited computational resources are very common.

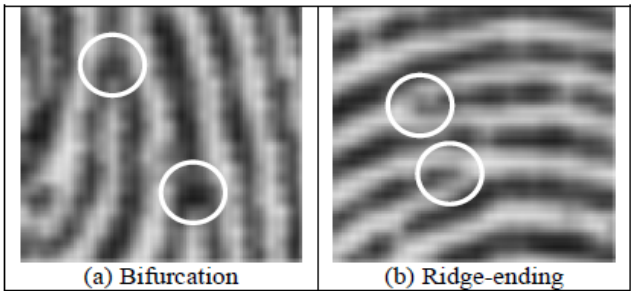

Fig 2.3: (a) Ridge Bifurcations (b) Ridge Endings

Minutiae, in fingerprint context, are the various ridge discontinuities of a fingerprint. There are more than 100

different types of minutiae have been identified, among which ridge bifurcations and endings (Fig.1.3) are the most widely used.

Minutiae-based methods have been used in many commercial fingerprint matching systems. Based primarily on a point pattern matching model, these methods rely heavily on the accuracy of minutiae extraction and the detection of landmarks like core and delta for pre-alignment. Spurious and missing minutiae can both introduce errors in minutiae correspondence. Equally problematic is the inability to detect landmarks to guide pre-alignment. Taken together, these problems lead to sub-optimal matching accuracy.

Fortunately, the contextual information provided by ridge flow and orientation in the neighborhood of detected minutiae can help eliminate spurious minutiae while compensating for the absence of genuinely missing minutiae both before and during matching. In addition, coupled with a core detection algorithm that can robustly handle missing or partially available landmarks for pre-alignment, significant improvement in matching accuracy can be expected. In this chapter, we will firstly review fingerprint feature extraction, minutiae representation, and registration, which are important components of fingerprint matching algorithms

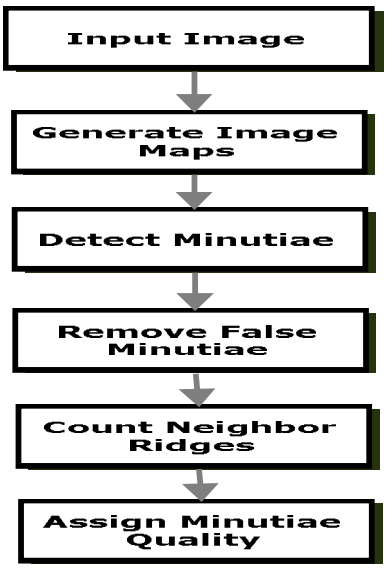

Fig 2.4: Minutiae Detection Process

A binary representation of the fingerprint is constructed by applying a rotated grid on the ridge flows of the fingerprint. Minutiae are generated by comparing each pixel neighborhood with a family of minutiae templates. Finally, a series of heuristic rules is used to merge and filter out the spurious minutiae (Fig.1.4).

Fingerprint verification determines whether two finger prints are impressions of a same finger or not. If fingerprints are from a same finger, their local ridge structures (minutia details) match each other topologically. Eighteen various types of local ridge descriptions have been identified. Ridge endings and ridge bifurcations (Fig. 1.5(a)) are the two most prominent structures which are usually called minutiae. Normally, minutiae detection algorithms firstly defines a set of template patterns that ridge endings or bifurcations would present like, and then compare local patterns with these template patterns while scanning the binary image of a fingerprint. With candidate minutiae indicated by the pattern matching stage, a post-process to remove false minutiae are performed with various criteria. Figure 1.5(b) displays detected minutiae on a skeleton finger print image.

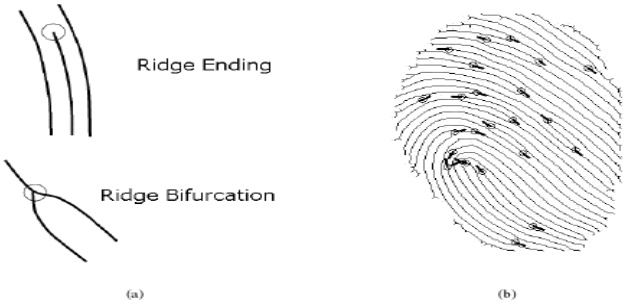

Fig 5.5: (a) Minutiae: Ridge Ending and Ridge Bifurcation (b) Detected Minutiae on a Skeleton Finger Print Image

3. Proposed Approach

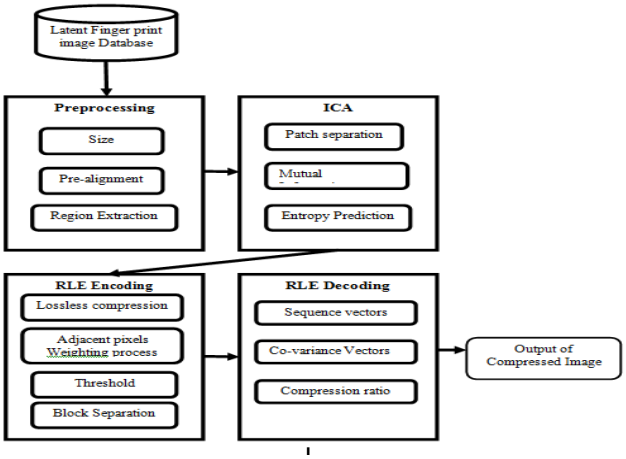

Fig 3.1: Architecture of System

Level 0: Construction of ICA Transformation Process

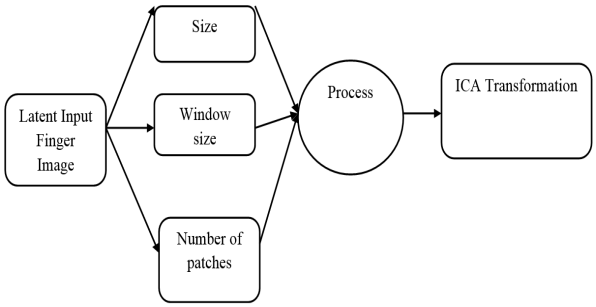

Fig 3.2: Construction of ICA Transformation Process

Level 1: RLE Encoding

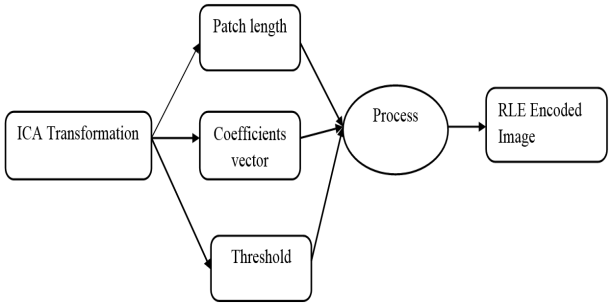

Fig 3.3: RLE Encoding

Level 2: RLE Decoding

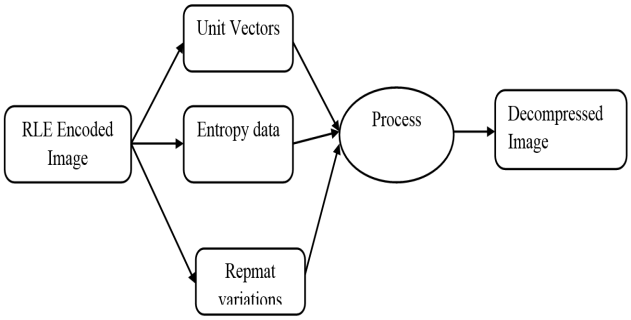

Fig 3.4: RLE Decoding

3.1 ICA Transformation

The ICA Transformation method uses a statistical "latent variables" model. Assume that we observe n linear mixtures $x_1, ..., x_n$ of n independent components

$Xj = a_{j1}s_1 + a_{j2}s_2 + \ldots + a_{jn}s_n$, for all j. (1)

In the ICA model, we assume that each mixture x_j as well as each independent component s_k is a random variable, instead of a proper time signal.

Matching Pursuit algorithm is normally use a full frame as a single block. It gives better compression without any blocking artifacts but it is not good for error resilience over noisy channels. Additively, computation is very heavy as MP is an iterative algorithm. To enhance error resilience capability along with reduced computational load, we have processed the image in blocks of 16x16 pixels. These blocks are encoded using variable number of coefficients until either of the stopping criteria is met which are minimum error threshold and maximum number of encoded coefficients.

The Latent fingerprint image sources have a lot of redundancy that is not removed by source coding. It is well known that removing redundancy by using transform coding before quantization generates much better codes. In transform coding, data is transformed to some other space where it has sparse representation with high peak and heavy tails.

For image extraction of basis functions, ICA has been used for incomplete, complete and over complete dictionaries. *tanh* non-linearity has been used for maximization of super-Gaussianity. For over complete dictionaries it utilizes the supposition of quasi-orthogonality. The algorithm is works in the following steps:

- Images patches are extracted from random locations from a group of images having same statistical characteristics.
- As a preprocessing step, they are processed by approximate orthogonalization pre-whitening.
- ICA algorithm is then used to extract the basis functions which are as independent as possible.
- Basis functions, extracted as independent sources, are not ordered. For better performance of entropy encoder, we need to order them in a fashion that gives a long trail of zeros in the end and most of nonzero coefficients (NZs) in the beginning of every block. As ICA basis functions are trained for a specific class, energy of each basis is a clear indication how much the specific basis function exists in that class. So, energy level is an appropriate criteria for ordering of basis functions to get better compression. This ordering gives excellent results for basis functions of both natural images and database images. Thus, in post-processing step, ICA bases are ordered by their energy level and then are normalized.

3.2 RLE Encoding Compression

This encoding method is frequently applied to graphics-type images (or pixels in a scan line) — simple compression algorithm in its own right.

RLE Approach is given below:

- Sequences of image elements X1, X2, …, Xn (Row by Row)
- Mapped to pairs (c1, l1), (c1, l2), …, (c1, ln)
- Where ci represent image intensity or colour and *li* the length of the ith run of pixels.
- (Not dissimilar to zero length suppression above)

The patches have been employed firstly to produce separate streams of DC coefficients (Direct Current. It'll define the basic shade for the whole block. The DC may also refer as constant component). AC coefficients (Alternating Components. The remaining coefficients are called the AC coefficients) and their indexes. The correlation among DC coefficients is exploited by using differential pulse code modulation (DPCM). Similarly indexes of the AC coefficients are also de-correlated by DPCM.

3.3 RLE Decoding Compression

The Run length decoding process is easy: If there aren't control pixel characters the coded symbol just corresponds to the original symbol, and if control pixel count occurred then it must be replaced with characters in a defined number of times. It can be noticed that the process of decoding image pixels. RLE algorithms are practically used in various image compression techniques like the well known BMP, PCX, TIFF, and is also used in PDF file format. Furthermore, RLE also exists as separate compression technique and there is also a file format called RLE (in various brands)

The proposed enhanced RLE algorithm

Step 1: Read the Latent Finger print image.
Step 2: Get the height N and the width M for the image
Step 3: Create an array, let it RLE (N, M), each element of this array consists of three fields for image channels.
Step 4: Convert the image to the main array; ICA (N,M).
Step 5: ICA (N, M) image to the main array; RLE (N,M).
Step 6: Let X=RLE (0,0); RLE (0,0) is the first element in an array.
Let TH=10, TH: the threshold.
Step 7: For I= 0 to N-1
Step 8: For J=0 to M-1
Step 9: If X-RLE (I,J) <= TH then
Let C=C+1
Else
Let X=RLE(I,J) and C=0

Step 10: End.

4. Experimental Results

In the results the effects of ICA Transform on fingerprint compression is studied. As we say in the phase one section, there are three different ways to construct the dictionaries. Here, the first is to randomly select 4096 patches and arrange them as columns of the dictionary (Random-SR in short). The second is to select patches according to orientations (Orientation-SR in short). In the phase two experiments, there are 1000 patches with window size 16×16. The third is to train the ICA by RLE method (Run Length Encoding). In the experiment, the test set is DATABASE 2 and the training set is DATABASE 1. For grey-level 8-bits per pixel images, the PSNR is computed as,

$$PSNR(I,I') = 10log_{10}\frac{255^2}{MSE(I,I')}(dB) \quad (2)$$

Table 1: Resulting Average of PSNR (dB) from Three Ways to build the Dictionary on Database

Selected Co-efficient	Random-SR	Orientation-SR	K-SVD-SR	RLE Method
1	20.27	20.26	20.36	21.58
2	23.84	23.84	24.34	25.12
3	24.91	24.90	25.72	27.22
4	26.4	26.39	30.99	31.85
5	26.97	26.95	32.77	32.99
6	27.88	27.86	29.55	32.08

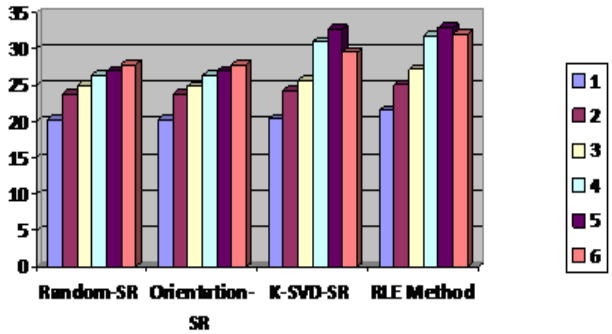

Fig 4.1: Comparison Chart

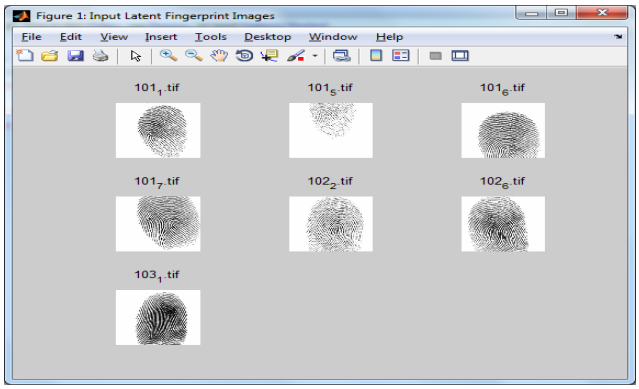

Fig 4.2: Input Latent Fingerprint Images

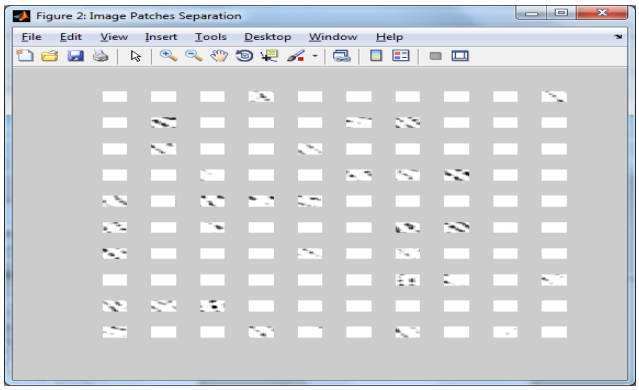

Fig 4.3: Image Patches Separation

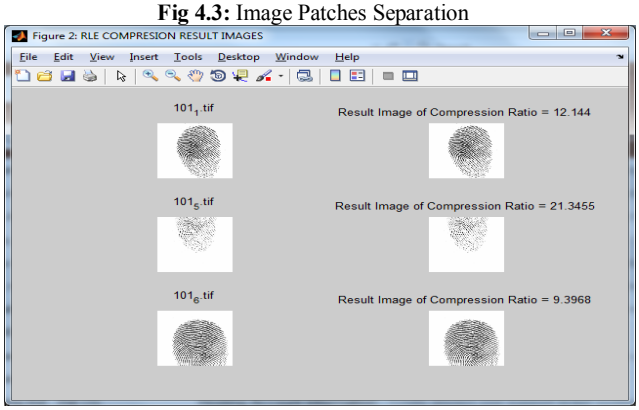

Fig 4.4: RLE Compression Result Images-1

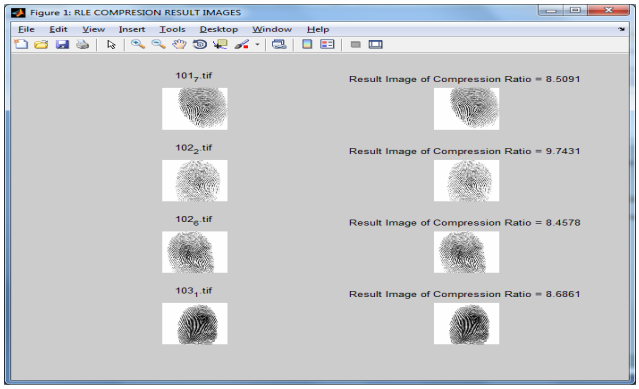

Fig 4.5: RLE Compression Result Images-2

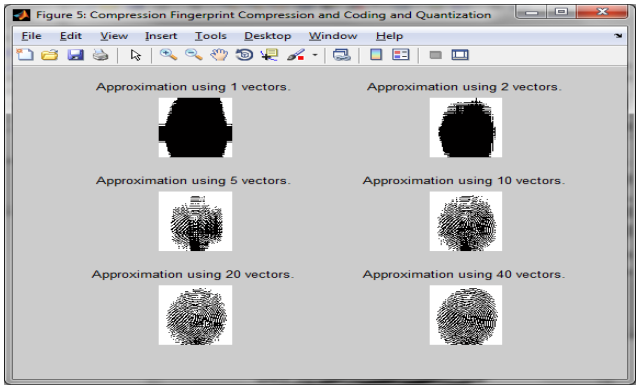

Fig 4.6: Fingerprint Compression Coding and Quantization

5. Conclusion & Future Scope

The algorithm proposed here is for lossless image compression ass it is evident from the algorithm, that the

exact image data (pixel values) are extracted from the compressed data stream without any loss. This is possible because the compression algorithm does not ignores or discards any original pixel value. Moreover the techniques such as approximate matching and run length encoding technique are intrinsically lossless.

This compression technique proves to be highly effective for images with large similar locality of pixel lay out. This technique will find extensive use in medical imaging sector because of its lossless characteristics and the medical images has large area of similar pixel layout pattern, like in latent images large area are black.

The algorithm of the proposed method is written in MATLAB programming language, such as in order that the algorithm can be used commercially. In future the X-Ray image compression system could be an extra option for processing in Latent images.

5. References

1. Maltoni D, Miao D, Jain AK, Prabhakar S, Handbook of Fingerprint Recognition, 2nd ed. London, U.K.: Springer-Verlag, 2009.
2. Ahmed N, Natarajan T, Rao KR. "Discrete cosine transform," IEEE Trans. Comput 1974; C-23(1):90-93.
3. Burrus CS, Gopinath RA, Guo H. Introduction to Wavelets and Wavelet Transforms: A Primer. Upper Saddle River, NJ, USA: Prentice-Hall, 1998.
4. Pennebaker W, Mitchell J. JPEG—Still Image Compression Standard. New York, NY, USA: Van Nostrand Reinhold, 1993.
5. Marcellin MW, Gormish MJ, Bilgin A, Boliek MP. "An overview of JPEG-2000," in Proc. IEEE Data Compress. Conf., Mar, 2000, 523-541.
6. Skodras, Christopoulos C, Ebrahimi T. "The JPEG 2000 still image compression standard," IEEE Signal Process. Mag 2001; 11(5):36-58.
7. Hopper T, Brislawn C, Bradley J. "WSQ gray-scale fingerprint image compression specification," Federal Bureau of Investigation, Criminal Justice Information Services, Washington, DC, USA, Tech. Rep. IAFIS-IC-0110-V2, Feb, 1993.
8. Brislawn CM, Bradley JN, Onyshczak RJ, Hopper T. "FBI compression standard for digitized fingerprint images," Proc. SPIE 1996; 2847:344-355.
9. Said, Pearlman WA. "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," IEEE Trans. Circuits Syst. Video Technol 1996; 6(3):243-250.
10. Sudhakar R, Karthiga R, Jayaraman S. "Fingerprint compression using contourlet transform with modified SPIHT algorithm," IJECE Iranian J. Electr. Comput. Eng 2005; 5(1):3-10.
11. Mallat SG, Zhang Z. "Matching pursuits with timefrequency dictionaries," IEEE Trans. Signal Process 1993; 41(12):3397-3415.
12. Chen SS, Donoho D, Saunders M. "Atomic decomposition by basis pursuit," SIAM Rev 2001; 43(1):129-159.
13. Wright J, Yang AY, Ganesh A, Sastry SS, Ma Y. "Robust face recognition via sparse representation," IEEE Trans. Pattern Anal. Mach. Intell 2009; 31(2):210-227.
14. Elad M, Aharon M. "Image denoising via sparse and redundant representation over learned dictionaries," IEEE Trans. Image Process 2006; 15(12):3736-3745.
15. Agarwal S, Roth D. "Learning a sparse representation for object detection," in Proc. Eur. Conf. Comput. Vis., 2002, 113-127.
16. Yang J, Wright J, Huang T, Ma Y. "Image super-resolution as sparse representation of raw image patches," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun, 2008, 1-8.
17. Skretting K, Engan K. "Image compression using learned dictionaries by RLS-DLA and compared with K-SVD," in Proc. IEEE ICASSP, May, 2011, 1517-1520.
18. Bryt O, Elad M. "Compression of facial images using the K-SVD algorithm," J. Vis. Commun. Image Represent 2008; 19(4):270-282.