International Journal of Computer Networks and Communications Security

VOL. 3, NO. 8, AUGUST 2015, 355–362
Available online at: www.ijcncs.org
E-ISSN 2308-9830 (Online) / ISSN 2410-0595 (Print)

An Efficient Fingerprint Compression Algorithm Using Sparse Coding

SHAHANAS K A[1] and SELIN M[2]

[1, 2] Department of Computer science and engineering, KMEA engineering college, Cochin, Kerala,India

E-mail: [1]shahanasamin@gmail.com, [2]snm.cs@kmeacollege.ac.in

ABSTRACT

Recognition of person by means of their biometric characteristics is very popular among the society. Among this, for personal identification fingerprint recognition is an important technology due to its unique structure. Human fingerprints are rich in details called minutiae. This can be used for identification marks for fingerprint verification. Large volume of fingerprint images are collected and stored day by day from a wider range of applications. Compression of data is commanding of under certain circumstances due to large amount of data transmission and efficient memory utilization. A new and efficient fingerprint compression algorithm using sparse representation is introduced. Obtaining an over complete dictionary from a set of fingerprint patches allows us to represent them as a sparse linear combination of dictionary atoms. In the algorithm, first construct a dictionary for predefined fingerprint image patches. For a new given fingerprint images, represent its patches according to the dictionary by computing l0- minimization and then quantize and encode the coefficients and other related information using lossless coding methods. A fast Fourier filtering transformation for image post processing helped to improve the contrast ratio of the regenerated image Here, considered the effect of various factors on compression results. In Automatic Fingerprint identification System, the main feature used to match two fingerprint images are minutiae. Therefore, the difference of the minutiae between pre- and post-compression is considered in the project. The experiments illustrate that the proposed algorithm is robust to extract minutiae. From the results, we can say that the proposed system provides better PSNR and verbosity for reconstructed images. Compression using sparse provides a better compression ratio also.

Keywords: *Compression; Dictionary learning; Fingerprint; Matching pursuit; Sparse Coding.*

1 INTRODUCTION

Personal identification is to associate a particular individual with an identity. It plays a critical role in our society. A wide variety of systems require reliable personal authentication schemes to either confirm or determine the identity of individuals requesting their services. In the absence of robust authentication schemes, these systems are vulnerable to the wiles of an impostor. Traditionally, passwords and ID cards have been used to restrict access to systems. The major advantages of this traditional personal identification are that:

(i) They are very simple

(ii) They can be easily integrated into different systems with a low cost.

However these approaches are not based on any inherent attributes of an individual to make a personal identification thus having number of disadvantages like tokens may be lost, stolen, forgotten, or misplaced PIN may be forgotten or guessed by impostors. Security can be easily breached in these systems when a password is divulged to an unauthorized user or a card is stolen by an impostor. Further, simple passwords are easy to guess by an impostor and difficult passwords may be hard to recall by a legitimate user. Therefore they are unable to satisfy the security requirements of our electronically interconnected information society. The emergence of biometrics has addressed the problems that plague traditional verification.

In the world of computer security, biometrics refers to authentication techniques that rely on measurable physiological and individual characteristics that can be automatically verified.

Among many biometric recognition technologies, fingerprint recognition is very popular for personal identification due to the uniqueness, universality, collectability and invariance. Typical fingerprint recognition methods employ feature-based image matching, where minutiae (i.e., ridge ending and ridge bifurcation) are extracted from the registered fingerprint image and the input fingerprint image, and the number of corresponding minutiae pairings between the two images is used to recognize a valid fingerprint image. The feature-based matching provides an effective way of identification for majority of people. The minutiae based automatic identification technique first locates the minutiae point and matches their relative placement in a given finger and the stored template.

Fingerprint compression using sparse representation has been implemented. Dictionary has been formulated from a set of fingerprint patches. First construct dictionary based on predefined fingerprint patches. Every fingerprint image is going to minimize by l0 minimization algorithm. Matching pursuit is a type of sparse approximation which involves finding the best matching projections of multidimensional data onto an overcomplete dictionary D.

Compared to general natural images, the fingerprint images have simpler structure and composed of ridges and valleys, in the local regions they look the same. So the whole images are sliced into square and non-overlapping patches. For these small patches, there are no problems about transformation and rotation. The size of the dictionary is not too large because the small blocks are smaller. The seemingly contradictive effect is achieved in a conventional optimization framework making use of gradient minimization, which can be probably controls how many non-zero gradients are resulted to approximate prominent structure in a structured sparsity management manner. Coefficients can be quantified by Lloyd's algorithm finding evenly spaced sets of points in subsets in to well saved and uniformly sized convex cells. Finally all values will be encoded using arithmetic encoding. It is a form of entropy encoding using lossless data compres-sion. It compact favorably with existing more sophistic algorithm especially at high compression ratios.

In proposed system, new approach has been proposed. That is sparse representation in which first initial dictionary has been constructed. In this, dictionary formation case with the algorithm has been implemented via singular valued decomposition by means of k-means clustering method. It works by iteratively altering between sparse encoding. The input data based on the current dictionary and updating a in the dictionary to better fit the data. Next every image will be compressed by orthogonal matching pursuit algorithm method l0 minimization process has been implemented. MP is a type of sparse approximation which involves finding the best matching projection of multidimensional data over complete dictionary D.

This paper is arranged as follows; section II summarizes the related works. Section III shows the details of fingerprint compression based on sparse coding. Section IV gives the experiment and at the end we draw a brief conclusion and the future work.

2 FINGERPRINT COMPRESSION USING SPARSE CODING

Sparse coding is the modeling of data vectors as sparse linear combinations of basic elements which is widely used in machine learning, neuroscience, signal processing, and statistics. It may focuses on the large-scale matrix factorization problem that consists of learning the basic set in order to adapt it to specific data. Variations of this problem include dictionary learning in signal processing. Here it proposes to address these tasks with a new online optimization algorithm, based on stochastic approximations, which scales up gracefully to large data sets with millions of training samples, and extends naturally to various matrix factorization formulations, making it suitable for a wide range of learning problems.

A. The Model And Algorithms – Sparse Coding

The model of sparse representation is given as, A = $[a_1, a_2, \ldots\ldots a_n] \in R^{M\times N}$, any new sample y $\in R^{M\times 1}$, is assumed to be represented as a linear combination of few columns from the dictionary A, as shown in equation 1. This is the only prior knowledge about the dictionary in our algorithm. Later, we will see the property can be ensured by constructing the dictionary properly.

$$y = Ax \qquad (1)$$

where $y \in R^{M\times 1}$, $A \in R^{M\times N}$ and $x = [x_1, x_2, \ldots\ldots x_n]^T \in R^{N\times 1}$. Obviously, the system $y = Ax$ is underdetermined when $M < N$. Therefore, its solution is not unique. According to the assumption, the representation is sparse. A proper solution can be obtained by solving the following optimization problem

$$(l^0) : \min ||x||_0 \quad \text{s.t. } Ax = y \qquad (2)$$

Solution of the optimization problem is expected to be very sparse, namely, $||x||0 << N$. The notation $||x||0$ counts the nonzero entries in x. Actually it is not a norm. However, without ambiguity, we still call it l0-norm. In fact, the compression of y can be achieved by compressing x. First, record the locations of its non-zero entries and their magnitudes. Second, quantize and encode the records. This is what we will do. Next, techniques for solving the optimization problem are given. Sparse solution by greedy algorithm is the first thought to solve the optimization problem l0 directly. However, the problem of finding the sparsest solution of the system is NP-hard [24]. The Matching Pursuit (MP), because of its simplicity and efficiency is often used to approximately solve the l0 problem. Many variants of the algorithm are available, offering improvements either in accuracy or in complexity.

Sparse Solution by l1-Minimization is a natural idea that the optimization problem which can be approximated by solving the following optimization problem

$$(l^p) : \min ||x||_p^p \quad \text{s.t. } Ax = y \qquad (3)$$

Obviously, the smaller p is, the closer the solutions of the two optimization problems $l^0$and l^p. This is because the magnitude of x is not important when p is very small. What does matter is whether x is equal to 0 or not. Therefore, p is theoretically chosen as small as possible. However, the optimization problem (3) is not convex if 0 < p < 1. It makes p = 1 the most ideal situation, namely, the following problems

$$(l^1) : \min ||x||_1 \quad \text{s.t. } Ax = y \qquad (4)$$

Recent developments in the field of sparse representation and compressed sensing [27] reveal that the solution of the optimization problem is approximately equal to the solution of the optimization problem if the optimal solution is sparse enough. The problem can be effectively solved by linear programming methods. In addition to the above algorithms, there are other algorithms for the optimization problems. There are also several well-developed software packages that handle this problem, which are freely shared on the web.

Algorithm 1 Fingerprint Compression Technique based on Sparse Representation [1].

```
1: For a given fingerprint, slice it into small Patches.
2: For each patch, its mean is calculated and subtracted from the patch.
3: For each patch, solve the l0 - minimization problem by OMP method.
4: Those coefficients whose absolute value is less than a given threshold are treated as zero. Record the remaining coefficients and their locations.
5: Encode the atom number of each patch, the mean value of each patch, and the indexes; quantize and encode the coefficients.
6: Output the compressed stream.
```

2.1 Dictionary Creation

The dictionary will be constructed in three different ways. First, we construct a training set. Then, the dictionary is obtained from the set. Choose the whole fingerprint images, and cut them into fixed-size square patches. Given these patches after the initial screening, a greedy algorithm is employed to construct the training samples.

- The first patch is added to the dictionary, which is initially empty.
- Then we check whether the next patch is sufficiently similar to all patches in the dictionary. If yes, the next patch is tested. Otherwise, the patch is added into the dictionary. Here, the similarity measure between two patches is calculated by solving the optimization problem.

$$s(p_1, p_2) = \min_t || \frac{p_1}{||p_1||_F^2} - t * \frac{p_2}{||p_2||_F^2} ||_F^2 \qquad (5)$$

where $||\bullet||_F^2$ is the Frobenius norm. p_1 and p_2 are the corresponding matrices of two patches. t, a parameter of the optimization problem (5), is a scaling factor.

- Repeat the second step until all patches have been tested.

Before the dictionary is constructed, the mean value of each patch is calculated and subtracted from the corresponding patch. Next, details of the three methods are given.

- The first method: choose fingerprint patches from the training samples at random and arrange these patches as columns of the dictionary matrix.
- The second method: in general, patches from foreground of a fingerprint have an orientation while the patches from the background don't have, as shown in Figure. 2. This fact can be used to construct the dictionary. Divide the interval [00, . . . , 1800] into equal-size intervals. Each interval is represented by an orientation. Choose the

same number of patches for each interval and arrange them into the dictionary.

- The third method: it is a training method called K-SVD. The dictionary is obtained by iteratively solving an optimization problem. Y is consisted of the training patches, A is the dictionary, X are the coefficients and Xi is the i th column of X. In the sparse solving stage, we compute the coefficients matrix X using MP method, which guarantees that the coefficient vector Xi has no more than T non-zero elements. Then, update each dictionary element based on the singular value decomposition (SVD).

$$\min_{A,X} ||Y - AX||_F^2 \text{ s.t } \forall_i, ||X_i||_0 < T \quad (6)$$

A novel algorithm for adapting dictionaries so as to represent signals sparsely are described below. Given a set of training signals ,the dictionary that leads to the best possible representations for each member in this set with strict sparsity constraints. Here uses the K-SVD algorithm that addresses the above task, generalizing the k-means algorithm. The K-SVD is an iterative method that alternates between sparse coding of the examples based on the current dictionary and an update process for the dictionary atoms so as to better fit the data. The update of the dictionary columns is done jointly with an update of the sparse representation coefficients related to it, resulting in accelerated convergence. The K-SVD algorithm is flexible and can work with any pursuit method, thereby tailoring the dictionary to the application in mind.

2.2 Compression of the fingerprint

Given a new fingerprint, slice it into square patches which have the same size with the training patches. The size of the patches has a direct impact on the compression efficiency. The algorithm becomes more efficient as the size increases. However, the computation complexity and the size of the dictionary also increase rapidly. The proper size should be chosen. In addition, to make the patches fit the dictionary better, the mean of each patch needs to be calculated and subtracted from the patch. After that, compute the sparse representation for each patch by solving the l_0 problem. Those coefficients whose absolute values are less than a given threshold are treated as zero. For each patch, four kinds of information need to be recorded. They are the mean value, the number about how many atoms to use, the coefficients and their locations. The tests show that many image patches require few coefficients. Consequently, compared with the use of a fixed number of coefficients, the method reduces the coding complexity and improves the compression ratio.

2.3 Entropy Coding and Quantization

Entropy coding of the atom number of each patch, the mean value of each patch, the coefficients and the indexes is carried out by static arithmetic coders. The atom number of each patch is separately coded. The mean value of each patch is also separately coded. The quantization of coefficients is performed using the Lloyd algorithm, learnt off-line from the coefficients which are obtained from the training set by the MP algorithm over the dictionary.

The first coefficient of each block is quantized with a larger number of bits than other coefficients and entropy-coded using a separate arithmetic coder. The model for the indexes is estimated by using the source statistics obtained off-line from the training set. The first index and other indexes are coded by the same arithmetic encoder. In the following experiments, the first coefficient is quantized with 6 bits and other coefficients are quantized with 4 bits.

2.4 Analysis of the Algorithm Complexity

The algorithm includes two parts, namely, the training process and the compression process. Because the training process is off-line, only the complexity of compression process is analyzed. Suppose the size of the patch is $m \times n$ and the number of patches in the dictionary is N. Each block is coded with L coefficients. L is the average number of non-zero elements in the coefficient vectors. To represent a patch with respect to the dictionary, each iteration of the MP algorithm includes mnN scalar products. The total number of scalar multiplications of each patch is $LmnN$. Given a whole fingerprint image with $M1 \times N1$ pixels. The number of patches of the fingerprint image is approximately equal to $M1{\times}N1/m{\times}n$ Therefore, the total number of scalar multiplications for compressing a fingerprint image is $M1{\times}N1/m{\times}n \times LmnN$, namely, $LM1N1N$.

Algorithm 1 summaries the complete compression process. The compressed stream doesn't include the dictionary and the information about the models. It consists solely of the encoding of the atom number of each patch, the mean value of each patch, the coefficients plus the indexes. In practice, only the compressed stream needs to be transmitted to restore the fingerprint. In both encoder and the decoder, the dictionary, the quantization tables of the coefficients and the statistic tables for arithmetic coding need to be stored.

Fingerprint database for fingerprint compression, which aim both to demonstrate the feasibility of the proposed compression algorithm and to validate the claims of the previous sections. First, the database used in this study is described. Then, the feasibility of the proposed method is proved.

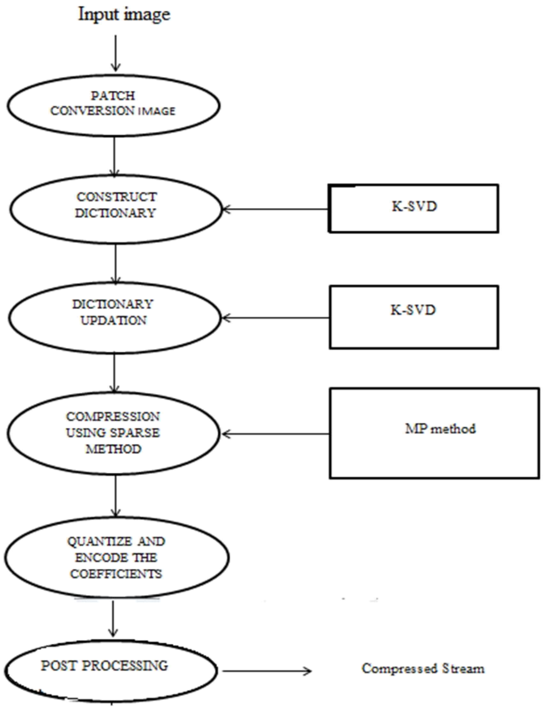


Fig. 1. Architecture of fingerprint compression using sparse

A. Databases

A set of training fingerprints including a major pattern types is used to construct a dictionary of fingerprint patches. The distribution of different pattern in this set and the distribution in the other database need not be similar they can be dissimilar also.

B. Feasibility of Fingerprint Compression on Sparse Representation

There are both a few large coefficients and other coefficients which are approximately equal to zero in the patches of fingerprints. For some patches they don't have coefficients at all. The mean values of the patches of database can be represented effectively from the background of the fingerprint images.

C. Experimental Results in Dictionary

The effects of different images in dictionary on fingerprint compression are here. Selecting number of patches at random and arranging them as columns of the dictionary, selecting patches as per orientations and training the dictionary by K-SVD method are the three different ways to construct the dictionary.

D. Choose the Algorithmic Parameters

The size of the patches and the size of the dictionary are the parameters in the compression based on sparse representation. The size of patches and compression efficiency are directly related. If the size is larger than the efficiency will be higher. If the dictionary is quite large the arbitrary patch can be represented well which causes more computational complexity. Only with experiments the size of the patch is chosen.

E. Comparison with Existing Fingerprint Compression Algorithms.

A comparison is made between the proposed methods with existing fingerprint compression algorithms. The same database is used to test sparse algorithm and all others. Matlab tool is used in the implementation of sparse compression algorithm. WSQ, JPEG, JPEG 2000 are the three images used in compression algorithms [1]. Normally, the fingerprint images are not the multiple of size of the patches. If not the left part of the fingerprint images is arranged into the columns vector if the same size as the patch in a given order. OMP algorithm can be used under the same dictionary can be used to represent the column vector. Each fingerprint image cannot be exactly compressed at the given compression ratios.

F. Design the Training Set

Consider many details of the images to target the problem of fingerprint. The number of training samples, components of the training set and how to modify the dictionary is taken into consideration.

1) The number of training samples:

As K-SVD method trains the dictionary, the effect of the number of training samples on the PSNR is to be considered.

2) Components of the training set:

General images are not easy to deal with even though the fingerprint images are simple. Some of the main feature of the fingerprints is in the orientation, ridge frequency and minutiae. These are important in the construction of dictionary.

3) Modify the dictionary :

Empirical information is to be used to modify the dictionary as it has the ability of learning. If we apply for longer period many patches cannot be represented well. So, these new patches and original patches can be used to update the dictionary.

G. Robustness

Gabor filtering does not rely on minutiae, but minutiae are mainly used in most Automatic Fingerprint Identification System to match two fingerprint images. So, we compare the difference of minutiae between pre and post compression in the illustration of the robustness of our algorithm. The accuracy rate, the recall rate and their sum are the important standard to measure robustness. Database with large collection of images is considered. Analysis is done on the basis of two; training and compression of the image. Compression using sparse provide better fidelity and quality to the regenerated image. Provide good compression ratio and PSNR value.

Data compression is used to quantify the reduction in data representation size. It is the ratio between the uncompressed and compressed size. Compression ratio using sparse is analyses to be of 100:1. Figure 5 shows the example of original image and regenerated image using sparse coding.

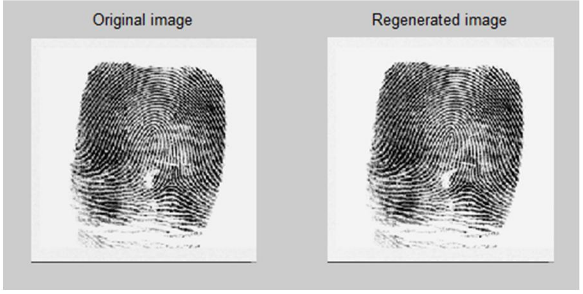


Fig. 2. sample image and regenerated image using sparse compression.

Another measure to compression is the PSNR value of compressed image. Which is the ratio between the maximum possible power of the signal and the power of the corrupted noise. PSNR is commonly used to measure the fidelity of the image. Acceptable values of PSNR for gray scale image is of from 15 to 30dB.

$$\text{PSNR} = 20log_{10}(\frac{MAX_f}{\sqrt{MSE}}) \qquad (7)$$

Where $\text{MSE} = \frac{1}{mn}\sum_0^{m-1}\sum_0^{n-1}||f(i,j) - g(i,j)||^2$

F: matrix data of original image. g: matrix data of degraded image, m: numbers of rows of pixels of the images and i represents the index of that row. n: represents the number of columns of pixels of the image and j represents the index of that column MAX_f is maximum signal value that exists in our original "known to be good" image.

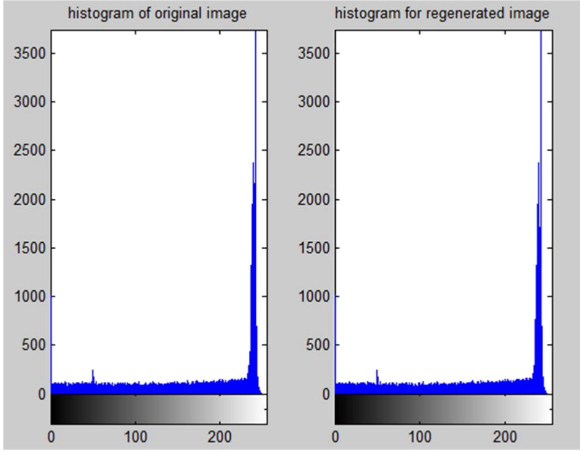


Fig. 3. Example Of histogram to demonstrates lossless compression of figure 2.

Contrast to noise is the measure for image quality of regenerated image.

$$C=\frac{|S_A - S_B|}{\sigma_0} \qquad (8)$$

S_A and S_B are signal intensities and σ_0 is the standard deviation of the pure image noise.

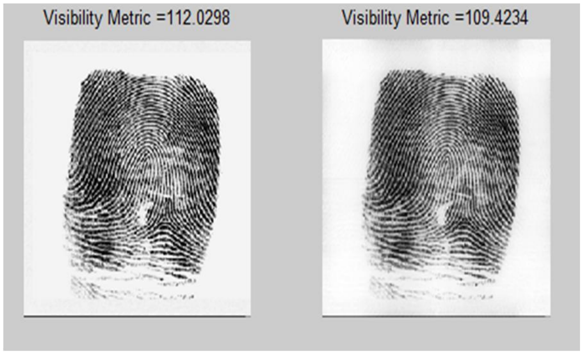


Fig. 4. shows the contrast to noise ratio of original and regenerated image of figure 2

Analysis of fingerprint compression using sparse is done through 2 phases. First is the training phase.

Figure 5 shows the different sample sizes and time to compute the training of sample set.
Second is the compression phase. Figure 6 shows the visibility measure of original and regenerated image, PSNR and compression ratios to analyses the compression complexity using sparse.

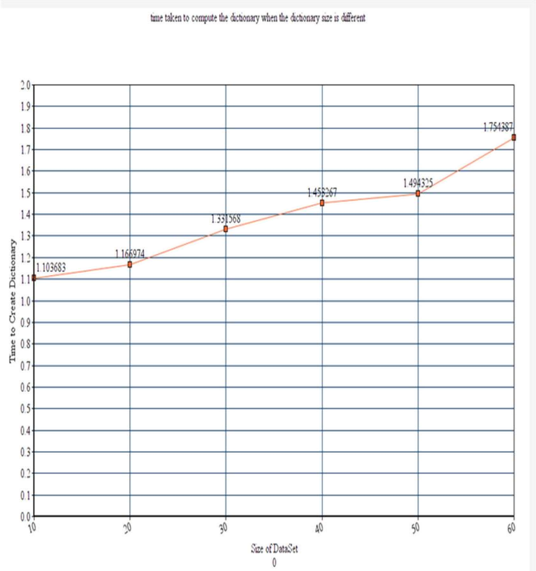
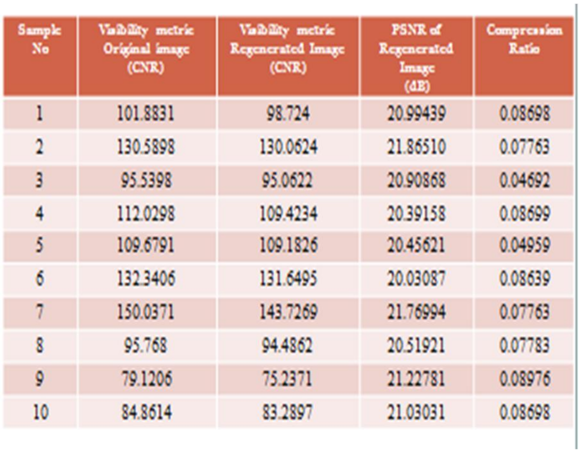


Fig. 5. line graph plots the time taken to compute the dictionary atoms with size of dataset



Sample No	Visibility metric Original image (CNR)	Visibility metric Regenerated Image (CNR)	PSNR of Regenerated Image (dB)	Compression Ratio
1	101.8831	98.724	20.99439	0.08698
2	130.5898	130.0624	21.86510	0.07763
3	95.5398	95.0622	20.90868	0.04692
4	112.0298	109.4234	20.39158	0.08699
5	109.6791	109.1826	20.45621	0.04959
6	132.3406	131.6495	20.03087	0.08639
7	150.0371	143.7269	21.76994	0.07763
8	95.768	94.4862	20.51921	0.07783
9	79.1206	75.2371	21.22781	0.08976
10	84.8614	83.2897	21.03031	0.08698

Fig. 6. Table shows the analysis results for compression phase of 10 random samples

3 CONCLUSION

A new compression algorithm adapted to fingerprint images is introduced. Despite the simplicity of proposed algorithms they compare favorably with existing more sophisticated algorithms, especially at high compression ratios. Due to the block-by-block processing mechanism, however, the algorithm has higher complexities.

The block effect of our fingerprint compression is less serious than that of JPEG. One of the main difficulties in developing compression algorithms for fingerprints resides in the need for preserving the fingerprint image. minutiae which are used in the identification. The algorithm can hold most of the minutiae robustly during the compression and reconstruction.

4 ACKNOWLEDGE

I am thankful to Mrs. Selin M, assistant professor of computers science and engineering department, KMEA kerala for her keen interest and guidance in my paper.

5 REFERENCES

[1] Guangqi Shao, Yanping Wu, Yong A, Xiao Liu, and Tiande Guo, Fingerprint compression based on sparse representation, Ieee Transactions On Image Processing, Vol. 23, No. 2, February 2014, pp. 489-501

[2] D. Maltoni, D. Miao, A. K. Jain, and S. Prabhakar, Handbook of Fingerprint recognition, 2nd ed. London, U.K.: Springer-Verlag, 2009.

[3] M. Elad and M. Aharon, “Image denoising via sparse and redundant representation over learned dictionaries,” IEEE Trans. Image Process., vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[4] W. Pennebaker and J. Mitchell, JPEG—Still Image Compression Standard. New York, NY, USA: Van Nostrand Reinhold, 1993.

[5] M. W. Marcellin, M. J. Gormish, A. Bilgin, and M. P. Boliek, “An overview of JPEG-2000,” in Proc. IEEE Data Compress. Conf., Mar. 2000, pp. 523–541.

[6] T. Hopper, C. Brislawn, and J. Bradley, “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.

[7] C. M. Brislawn, J. N. Bradley, R. J. Onyshczak, and T. Hopper, “FBI compression standard for digitized fingerprint images,” Proc. SPIE, vol. 2847, pp. 344–355, Aug. 1996. An Efficient Fingerprint Compression Using Sparse Representation

[8] A. Said and W. A. Pearlman, “A new, fast, and efficient image codec based on set partitioning in hierarchical trees,” IEEE Trans. Circuits Syst. Video Technol., vol. 6, no. 3, pp. 243–250, Jun. 1996.

[9] C. S. Burrus, R. A. Gopinath, and H. Guo, Introduction to Wavelets and Wavelet Transforms: A Primer. Upper Saddle River, NJ, USA: Prentice- Hall, 1998.

[10] O. Bryt and M. Elad, "Compression of facial images using the K-SVD algorithm," J. Vis. Commun. Image Represent., vol. 19, no. 4, pp. 270–282, 2008.

[11] A. J. Ferreira and M. A. T. Figueiredo, "Class-adapted image compression using independent component analysis," in Proc. Int. Conf. Image Process., vol. 1. 2003, pp. 625–628.

[12] R. Sudhakar, R. Karthiga, and S. Jayaraman, "Fingerprint compression using contourlet transform with modified SPIHT algorithm," IJECE Iranian J. Electr. Comput. Eng., vol. 5, no. 1, pp. 3–10, 2005.

[13] S. G. Mallat and Z. Zhang, "Matching pursuits with timefrequency dictionaries," IEEE Trans. Signal Process., vol. 41, no. 12, pp. 3397–3415, Dec. 1993.

[14] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard," IEEE Signal Process. Mag., vol. 11, no. 5, pp. 36–58, Sep. 2001.

[15] S. S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," SIAM Rev., vol. 43, no. 1, pp. 129–159, 2001. An Efficient Fingerprint Compression Using Sparse Representation

[16] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 2, pp. 210–227, Feb. 2009.

[17] S. Agarwal and D. Roth, "Learning a sparse representation for object detection," in Proc. Eur. Conf. Comput. Vis., 2002, pp. 113–127.

[18] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in Proc. Conf. Rec. 27th Asilomar Conf. Signals, Syst. Comput., Nov. 1993, pp. 40–44

[19] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution as sparse representation of raw image patches," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2008, pp. 1–8.

[20] K. Skretting and K. Engan, "Image compression using learned dictionaries by RLS-DLA and compared with K-SVD," in Proc. IEEE ICASSP, May 2011, pp. 1517–1520.

[21] A. J. Ferreira and M. A. T. Figueiredo, "On the use of independent component analysis for image compression," Signal Process., ImageCommun., vol. 21, no. 5, pp. 378–389, 2006.

[22] P. Paatero and U. Tapper, "Positive matrix factorization: A nonnegative factor model with optimal utilization of error estimates of data values," Environmetrics, vol. 5, no. 1, pp. 111–126, 1994.

[23] D. D. Leeand and H. S. Seung, "Learning the parts of objects by nonnegative matrix factorization," Nature, vol. 401, pp. 799–791, Oct. 1999. An Efficient Fingerprint Compression Using Sparse Representation

[24] E. Amaldi and V. Kann, "On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems," Theoretical Comput. Sci., vol. 209, nos. 1–2, pp. 237–260, 1998.

[25] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," IEEE Trans. Signal Process., vol. 41, no. 12, pp. 3397–3415, Dec. 1993.

[26] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," IEEE Trans. Comput., vol. C-23, no. 1, pp. 90–93, Jan. 1974.

[27] Y. Y. Zhou, T. D. Guo, and M. Wu, "Fingerprint image compression algorithm based on matrix optimization," in Proc. 6th Int. Conf. Digital Content, Multimedia Technol. Appl., 2010, pp. 14–19.

[28] M. Aharon, M. Elad, and A. M. Bruckstein, "The K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representation," IEEE Trans. Signal Process., vol. 54, pp. 4311–4322, 2006.

[29] R. Sudhakar, R. Karthiga, and S. Jayaraman, "Fingerprint compression using contourlet transform with modified SPIHT algorithm," IJECE Iranian J. Electr. Comput. Eng., vol. 5, no. 1, pp. 3–10, 2005.