International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 11, November 2015
Available at http://internationaljournalofresearch.org

Fingerprint Compression Based on Sparse Representation

Mr.D.Shiva Rama Krishna [1], D Mahesh [2]

[1] Assistant Professor, Dept of CSE, Marri Laxman Reddy Institute Of Technology & Management, Hyderabad, Telangana

[2] M.Tech CSE (PG Scholar), Dept of CSE, Marri Laxman Reddy Institute Of Technology & Management, Hyderabad, Telangana

Abstract—

A new fingerprint compression algorithm based on sparse representation is introduced. Obtaining an overcomplete dictionary from a set of fingerprint patches allows us to repre-sent them as a sparse linear combination of dictionary atoms. In the algorithm, we first construct a dictionary for predefined fingerprint image patches. For a new given fingerprint images, represent its patches according to the dictionary by computing l^0-minimization and then quantize and encode the representation. In this paper, we consider the effect of various factors on com-pression results. Three groups of fingerprint images are tested. The experiments demonstrate that our algorithm is efficient compared with several competing compression techniques (JPEG, JPEG 2000, and WSQ), especially at high compression ratios. The experiments also illustrate that the proposed algorithm is robust to extract minutiae.

Index Terms— Fingerprint; compression; sparse representa-tion; JPEG 2000; JPEG; WSQ; PSNR

I. INTRODUCTION

RECOGNITION of persons by means of biometric char-acteristics is an important technology in the society, because biometric identifiers can't be shared and they intrinsi-cally 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 consume the amount of memory. Fingerprint image compression is a key technique to solve the problem.

Generally, compression technologies can be classed into lossless and lossy.

Lossless compression allows the exact original images to be reconstructed from the compressed data. Lossless compression technologies are used in cases where it is important that the original and the decompressed data are identical. Avoiding distortion limits their compression efficiency. When used in image compression where slight distortion is acceptable, lossless compression technologies are often employed in the output coefficients of lossy compression.

Lossy compression technologies usually transform an image into another domain, quantize and encode its coefficients. During the last three decades, transform-based image com-pression technologies have been extensively researched and some standards have appeared. Two most common options of transformation are the Discrete Cosine Transform (DCT) [2] and the Discrete Wavelet Transform (DWT) [3].

The DCT-based encoder can be thought of as compression of a stream of 8 × 8 small block of images. This transform has been adopted in JPEG [4]. The JPEG compression scheme has many advantages such as simplicity, universality and availability. However, it has a bad performance at low bit-rates mainly because of the underlying block-based DCT scheme. For this reason, as early as 1995, the JPEG-committee began to develop a new wavelet-based compression standard for still images, namely JPEG 2000 [5], [6]. The DWT-based algo-rithms include three steps: a DWT computation of the normal-ized image, quantization of the DWT coefficients and lossless coding of the quantized coefficients. The detail can be found in [7] and [8]. Compared with JPEG, JPEG 2000 provides many features that support scalable and interactive access to large-sized image. It also allows extraction of different

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 11, November 2015
Available at http://internationaljournalofresearch.org

resolutions, pixel fidelities, regions of interest, components and etc. There are several other DWT-based algorithms, such as Set Partitioning in Hierarchical Trees (SPIHT) Algorithm [9].

The above algorithms are for general image compression. Targeted at fingerprint images, there are special compression algorithms. The most common is Wavelet Scalar Quantization (WSQ). It became the FBI standard for the compression of 500 dpi fingerprint images [7]. Inspired by the WSQ algorithm, a few wavelet packet based fingerprint compression schemes have been developed. In addition to WSQ, there are other algorithms for fingerprint compression, such as Contourlet Transform (CT) [10].

These algorithms have a common shortcoming, namely, without the ability of learning. The fingerprint images can't be compressed well now. They will not be compressed well later. In this paper, a novel approach based on sparse representation is given. The proposed method has the ability by updating the dictionary. The detail can be found in Section IV and V. The specific process is as follows: construct a base matrix whose columns represent features of the fingerprint images, referring the matrix dictionary whose columns are called atoms; for a given whole fingerprint, divide it into small blocks called patches whose number of pixels are equal to the dimension of the atoms; use the method of sparse representation to obtain the coefficients; then, quantize the coefficients; last, encode the coefficients and other related information using lossless coding methods.

In most instances, the evaluation of compression perfor-mance of the algorithms is restricted to Peak Signal to Noise Ratio (PSNR) computation. The effects on actual fingerprint matching or recognition are not investigated. In this paper, we will take it into consideration. In most Automatic Fingerprint identification System (AFIS), the main feature used to match two fingerprint images are minutiae (ridges endings and bifur-cations). Therefore, the difference of the minutiae between pre- and post-compression is considered in the paper.

This paper is arranged as follows: section II summarizes the related works and gives some thoughts on the sparse representation; the model of the sparse representation and the algorithms for the model are set up in section III; the details of fingerprint compression based on sparse representation is given in section IV; experiments will be given in section V; in the end, we draw a brief conclusion and the future work.

II. RELATED WORKS AND SOME THOUGHTS

The field of sparse representation is relatively young. Early signs of its core ideas appeared in a pioneering work [11]. In that paper, the authors introduced the concept of dictionaries and put forward some of the core ideas which later became essential in the field such as a greedy pursuit technique. Thereafter, S. S. Chen, D. Donoho and M. Saunders [12] introduced another pursuit technique which used l^1-norm for sparse. It is surprising that the proper solution often could be obtained by solving a convex programming task. Since the two seminal works, researchers have contributed a great deal in the field. The activity in this field is spread over various disciplines. There are already many successful applications in various fields, such as face recognition [13], image denois-ing [14], object detection [15] and super-resolution image reconstruction [16].

In paper [13], the authors proposed a general classification algorithm for object recognition based on a sparse representa-tion computed by l^1-minimization. On one hand, the algorithm based on sparse representation has a better performance than other algorithms such as nearest neighbor, nearest subspace and linear SVM; on the other hand, the new framework pro-vided new insights into face recognition: with sparsity properly harnessed, the choice of features becomes less important than the number of features. Indeed, this phenomenon is common in the fields of sparse representation. It doesn't only exist in the face recognition, but also appears in other situations.

In paper [14], based on sparse and redundant representations on over-complete dictionary, the authors designed an algorithm that could remove the zero-mean white and homogeneous Gaussian additive noise from a given image. In this paper, we can see that the content of the dictionary is of importance. The importance is embodied in two aspects. On one hand, the dictionary should correctly reflect the content of the images; on the other hand, the dictionary is large enough that the given image can be represented sparsely. These two points are absolutely vital for the methods based on sparse representation.

Sparse representation has already some applications in image compression [17], [18]. In paper [17], the

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 11, November 2015
Available at http://internationaljournalofresearch.org

experiments show that the proposed algorithm has good performance. However, its compression efficiency is consistently lower than JPEG 2000's. If more general natural images are tested, this phenomenon will be more obvious that the compres-sion efficiency is lower than the state-of-the-art compression technologies. In paper [18], the experiments show success compared to several known compression techniques. However, the authors emphasize that an essential pre-process stage for this method is an image alignment procedure. It is hard to do in the practical application.

There are other algorithms [19]–[21] for fingerprint image compression under a linear model assumption. In paper [20], [21], the authors showed how to exploit the data-dependent nature of Independent Component Analysis (ICA) to compression special images (face and fingerprint images). The experiments of the two papers suggested that, for special class, it was not worth to use over-complete dictionaries. In this paper, we show the fingerprint images can be compressed better under an over-complete dictionary if it is properly constructed. In paper [19], the authors proposed an algo-rithm of fingerprint compression based on Nonnegative Matrix Factorization (NMF) [22], [23]. Although NMF has some successful applications, it also has shortcomings. In some cases, non-negativity is not necessary. For example, in the image compression, what is considered is how to reduce the difference between pre- and post-compression rather than non-negativity.

Besides, we think the methods based on sparse representa-tion don't work very well in the general image compression field. The reasons are as follows: the contents of the general images are so rich that there is no proper dictionary under which the given image can be represented sparsely; even if there is one, the size of the dictionary may be too large to be computed effectively. For example, the deformation, rotation, translation and the noise all can make the dictionary become too large. Therefore, sparse representation should be employed in special image compression field in which there are no above shortcomings. The field of fingerprint image compression is one of them. The details can be found in section IV and V, where we will give the reasons and the accounts for the feasibility of fingerprint compression based on sparse representation.

III. THE MODEL AND ALGORITHMS OF SPARSE REPRESENTATION

A. The Model of Sparse Representation

Given $A = [a_1, a_2, \ldots, a_N] \in R^{M \times N}$, any new sample $y \in R^{M \times 1}$, is assumed to be represented as a linear combination of

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 11, November 2015
Available at http://internationaljournalofresearch.org

few columns from the dictionary A, as shown in formula (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.

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

where $y \in R^{M\times 1}$, $A \in R^{M\times N}$ and $x = [x_1, x_2, \ldots, x_N]^T \in R^{N\times 1}$.

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:

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

Solution of the optimization problem is expected to be very sparse, namely, $\|x\|_0 \ll N$. The notation $\|x\|_0$ counts the non-zero entries in x. Actually it is not a norm. However, without ambiguity, we still call it l^0-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.

B. Sparse Solution by Greedy Algorithm

Researchers' first thought is to solve the optimization problem l^0 directly. However, the problem of finding the sparsest solution of the system (2) is NP-hard [24]. The Matching Pursuit (MP) [25] because of its simplicity and efficiency is often used to approximately solve the l^0 problem. Many variants of the algorithm are available, offering improvements either in accuracy or/and in complexity. Although the theoretical analysis of these algorithms is difficult, experiments show that they behave quite well when the number of non-zero entries is low.

C. Sparse Solution by l^1-Minimization

It is a natural idea that the optimization problem (2) can be approximated by solving the following optimization problem:

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

where $p > 0$ and $\|x\|_p^p = \sum_{i=1}^{n} |x_i|^p$.

Obviously, the smaller p is, the closer the solutions of the two optimization problems l^0 and l^p are, as illustrated in Fig. 1. 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.

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

Recent developments in the field of sparse representation

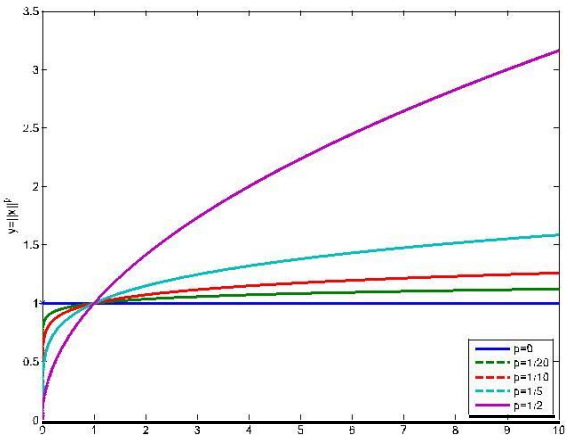


Fig. 1. The behavior of $\|x\|_p^p$ for various values of p. As p tends to zero, $\|x\|_p^p$ approaches the l^0-norm.

is sparse enough. The problem (4) can be effectively solved by linear programming methods.

In addition to the above algorithms, there are other algorithms [31], [32] for the problems (2) or (4). There are also several well-developed software packages that handle this problem, which are freely shared on the web. These include l^1-magic by Candes and Romberg, Sparselab managed by David Donoho, SparCo by Michael Friedlander and others.

IV. Fingerprint Compression Based on Representation

In this section, we give the details about how to use sparse representation to compress fingerprint images. The part includes construction of the dictionary, compression of a given fingerprint, quantization and coding and analysis of the algorithm complexity.

In the preceding paragraphs, it is mentioned that the size of the dictionary may be too large when it contains as much information as possible. Therefore, to obtain a dictionary with a modest size, the preprocessing is indispensable. Influenced by transformation, rotation and noise, the fingerprints of the same finger may look very different. What we first think is that each fingerprint image is pre-aligned, independently of the others. The most common pre-alignment technique is to translate and rotate the fingerprint according to the position of the core point. Unfortunately, reliable detection of the core is very difficult in fingerprint images with poor quality. Even if the core is correctly detected, the size of the dictionary may be overlarge because the size of a whole fingerprint image is too large.

Compared with general natural images, the fingerprint images have simpler structure. They are only composed of ridges and valleys. In the local regions, they look the same. Therefore, to solve these two problems, the whole image is sliced into square and non-overlapping small patches. For

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 11, November 2015
Available at http://internationaljournalofresearch.org

and compressed sensing [26]–[30] reveal that the solution of the optimization problem (4) is approximately equal to the solution of the optimization problem (2) if the optimal solution

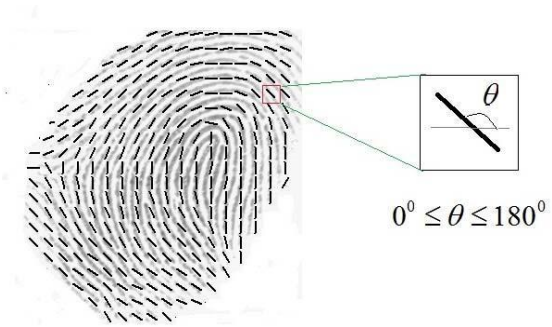


Fig. 2. A fingerprint image with its corresponding orientation image computed over a square-meshed grid. Each element denotes the local orientation of the fingerprint ridges.

A. Construction of the Dictionary

In this paper, the dictionary will be constructed in three ways. First, we construct a training set. Then, the dictionary is obtained from the set. Choose the whole fingerprint images, 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 dic-tionary. Here, the similarity measure between two patches is calculated by solving the optimization problem (5).

$$S(P_1, P_2) = \min_t \left\| \frac{P_1}{\|P_1\|_F} - t * \frac{P_2}{\|P_2\|_F} \right\|_F^2 \quad (5)$$

where $\|\bullet\|_F$ 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 Fig. 2. This fact can be used to construct the dictionary. Divide the interval $[0^0, \ldots, 180^0]$ into equal-size intervals. Each interval is represented by an orientation (the middle value of each interval is chosen). 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 [33], [34]. The dictionary is obtained by itera-tively solving an optimization problem (6). Y is consisted of the training patches, A is the dictionary, X are the coefficients and X_i is the *ith* column of X . In the sparse solving stage, we compute the coefficients matrix X

these small patches, there are no problems about transforma-tion and rotation. The size of the dictionary is not too large because the small blocks are relatively smaller.

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 11, November 2015
Available at http://internationaljournalofresearch.org

using MP method, which guarantees that the coefficient vector X_i has no more than T non-zero elements. Then, update each dictionary element based on the singular value decomposition (SVD).

$$\min_{A, X} \|Y - AX\|_2 \quad s.t. \quad \forall i, \|X_i\|_0 < T \qquad (6)$$

In the following experiment, the three kinds of dictionaries will be compared.

B. Compression of a Given 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. How to choose this size will be given in Section V. 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.

C. 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 [35]. 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 [36], 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.

D. 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

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 11, November 2015
Available at http://internationaljournalofresearch.org

Algorithm 1 Fingerprint Compression Based on Sparse Representation

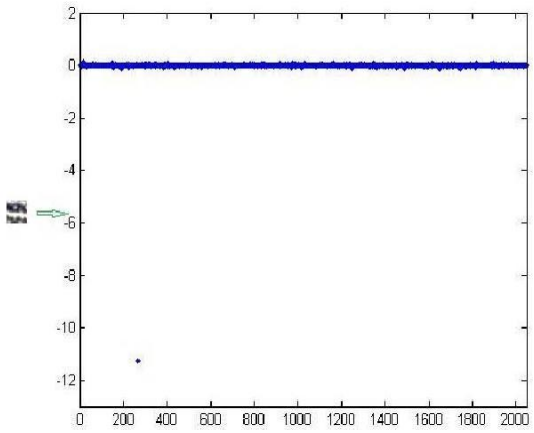
1: For a given fingerprint, slice it into small patches.
2: For each patch, its mean is calculated and subtracted from the patch.
3: For each patch, solve the l^0-minimization problem by MP method.
4: Those coefficients whose absolute value are less than a given threshold are treated as zero. Record 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.

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 $M_1 \times N_1$ pixels. The number of patches of the fingerprint image is approximately equal to $\frac{M_1 \times N_1}{m \times n}$. Therefore, the total number of scalar multiplications for compressing a fingerprint image is $\times$ *LmnN*, namely, $L\, M_1\, N_1\, N$.

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. In our experiments, this leads to less than 6 Mbytes. The compression rate equals the ratio of the size of original image and that of the com-pressed stream.

V. Experiments

In this section, we present experiments on 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 and experimental results on three different dictionaries is given. Next, How to choose proper parameters is studied. After that, we compare our method with existing fingerprint



compression algorithms, such as WSQ, JPEG, and JPEG 2000. Then, we study an important issue on sparse representation, namely, how to design the training set. Finally, through the experiment, it is shown the proposed compression algorithm is robust to extract minutiae.

A. Databases

To construct a dictionary of fingerprint patches, we use a set of training fingerprints which includes the major pattern

Fig. 3. The representation of a patch with only one large coefficient. The small image on the left is a patch with 20 × 20 size. The graph on the right is the patch's representation by l_1 -minimization under a dictionary with 400 × 2048 size.

types. The distribution of different pattern in this set is not necessarily similar to the distribution in the other database. There are 5 groups of fingerprint images (referred to as DATABASE 1, DATABASE 2, DATABASE 3, DATABASE 4 and DATABASE 5) in the experiments.

- DATABASE 1: 18 fingerprints, which are used as training set.
- DATABASE 2: 30 fingerprints that are used to choose proper parameters.
- DATABASE 3: 50 fingerprints that are used to compare our method with existing fingerprint compression algorithms and to test the robustness of our algorithm to extract minutiae.
- DATABASE 4: the public fingerprint database, FVC2002_DB3_B, including 80 fingerprints with size 300 × 300, which are used to compare our method with existing fingerprint compression algorithms.
- DATABASE 5: the public fingerprint database, FVC2004_DB2_B, including 80 fingerprints with size 328 × 364, which are also used to compare our method with existing fingerprint compression algorithms.

In the above five groups of fingerprint images, the first three groups come from the same database. The size of the

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 11, November 2015
Available at http://internationaljournalofresearch.org

fingerprints in the database is 640 × 640. The quality of fingerprints in the database is good. The three groups of fingerprint images can be downloaded from [37]. To show the dictionary trained from DATABASE 1 is universal for fingerprint images, DATABASE 4 and DATABASE 5 are also tested.

B. The Feasibility of Fingerprint Compression Based on Sparse Representation

In this part, we show the patches of fingerprints really have a sparse representation. See Fig. 4(a), 4(b) and 4(c), there are only a few large coefficients, while other coefficients are approximately equal to zero. More amazedly, there is only one large coefficient for many patches, an example shown in Fig. 3. In some special cases, there is no coefficient for some patches, for example, most of the patches from the background

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 11, November 2015
Available at http://internationaljournalofresearch.org

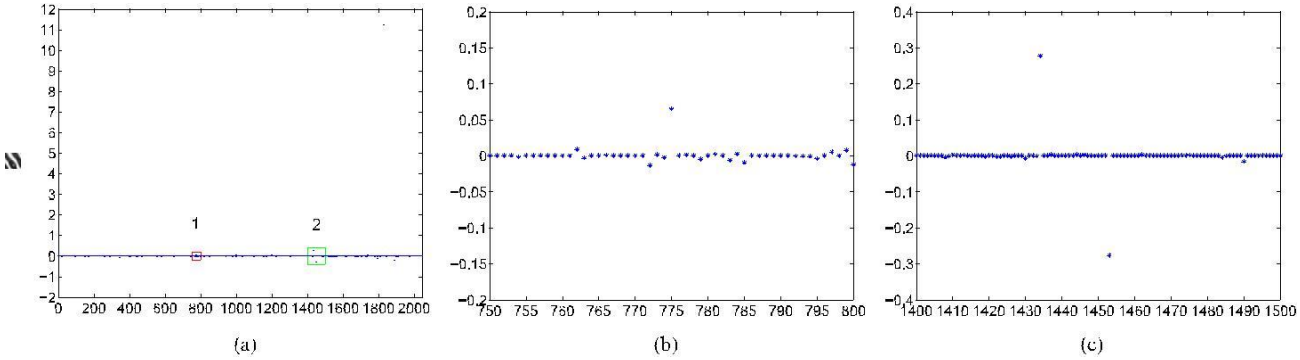


Fig. 4. The representation of a patch with 20 × 20 size. 4(a) A patch and its representation by l_1 -minimization under a dictionary. 4(b) The enlarged image of the block 1 showed in Fig. 4(a). 4(c) The enlarged image of the block 2 showed in Fig. 4(a).

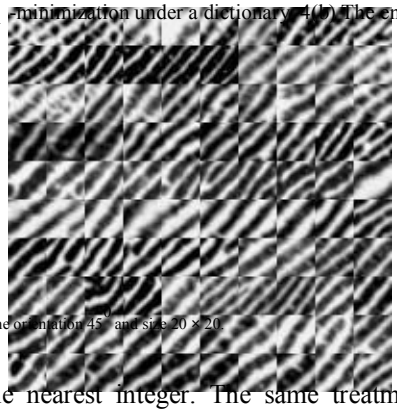


Fig. 5. 100 patches with the orientation 45° and size 20 × 20.

of the fingerprint images in DATABASE 1, DATABASE 2 and DATABASE 3. It is good enough to represent these patches using their mean values. These experiments show that the patches can be represented as a linear combination of few columns. The fact makes the proposed fingerprint compression method feasible.

C. Experimental Results on Different Dictionaries

In this section, the effects of different dictionaries on fingerprint compression is studied. As we say in the pre-vious 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 accord-ing to orientations (Orientation-SR in short). In the experi-ments, there are 8 orientations. See Fig. 5, they are 100 patches with orientation 45^0 and size 20 × 20. The third is to train the dictionary by K-SVD method (K-SVD-SR in short).

In the experiment, the test set is DATABASE 2 and the training set is DATABASE 1. The size of dictionary is 144 × 4096. For grey-level 8-bits per pixel images, the PSNR is computed as

it to the nearest integer. The same treatment is adopted in the following experiments. These results show that the

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

With Mean Square Error (MSE) defined as

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 11, November 2015
Available at http://internationaljournalofresearch.org

$$MSE(I, I^{-}) = \frac{1}{M^{-} \times N^{-}} \sum_{i=1}^{N^{-}} \sum_{j=1}^{M^{-}} (I(i,j) - I^{-}(i,j))^2 \quad (8)$$

where M^{-} and N^{-} are the lengths of fingerprint, I and I^{-} are the intensities of the original and reconstructed images. Table I and Fig. 6 shows the performance of the proposed algorithm under different dictionaries. The horizontal axis in Fig. 6 represents the percentage of selected coefficients for each patch and the vertical axis represents the mean values of the PSNR. If the number of the selected coefficients corresponding to the given percentage is not an integer, round

K-SVD-SR outperforms two other methods. An additional experiment shows the performance of the K-SVD-SR under the dictionary with 144 × 4096 is still better than that of a dictionary obtained by directly arranging all the patches (26843 patches). The mean values of the PSNR under the dictionary with 144 × 26843 are 21.25*dB*, 27.75*dB*, 29.71*dB* and 30.95*dB* at 1, 4, 7 and 10 percent of coefficients, respec-tively. It is concluded that proper training method is necessary and how to construct the dictionary is important. Fig. 6 also shows the Orientation-SR almost has the same performance as the Random-SR. Without the process of training, whether the patches are elaborately chosen makes little difference. In the following experiments, the dictionary is constructed in the third way.

D. How to Choose the Algorithmic Parameters

In the section, a study of the algorithmic parameters is conducted. In the experiment, the training set is DATABASE 1 and the test set is DATABASE 2.

For the compression based on sparse representation, the most important parameters are the size of patches and the size of the dictionary. The size of patches has a direct on compression efficiency. The larger the size is, the higher the efficiency is. However, to represent an arbitrary patch well, the size of the dictionary needs to be sufficiently large. This causes

(7) more computational complexity. So far, there is no good way to estimate the parameter. In the paper, the size of patches is

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 11, November 2015
Available at http://internationaljournalofresearch.org

TABLE I
RESULTING AVERAGE OF PSNR (DB) FROM THREE WAYS TO BUILD THE DICTIONARY ON DATABASE 3

Percentage of Selected Coefficients	1	2	3	4	5	6	7	8	9	10
Random-SR	20.27	23.84	24.91	26.4	26.97	27.88	28.26	28.91	29.20	29.46
Orientation-SR	20.26	23.84	24.90	26.39	26.95	27.86	28.23	28.88	29.17	29.42
K-SVD-SR	20.36	24.34	25.72	27.95	28.87	30.46	31.15	32.34	32.86	33.34

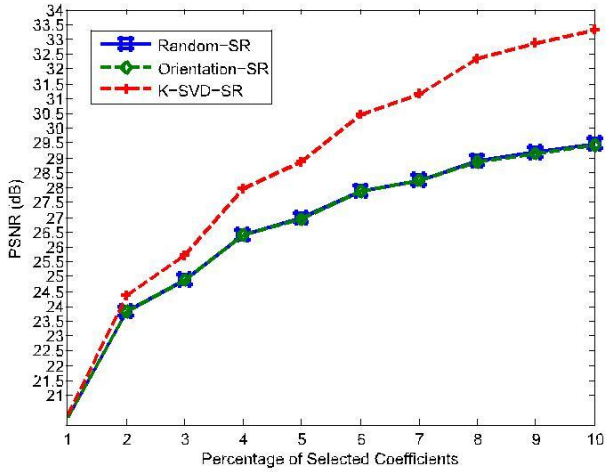


Fig. 6. Average performance of the proposed algorithms under different dictionaries, for various percentages of selected coefficients.

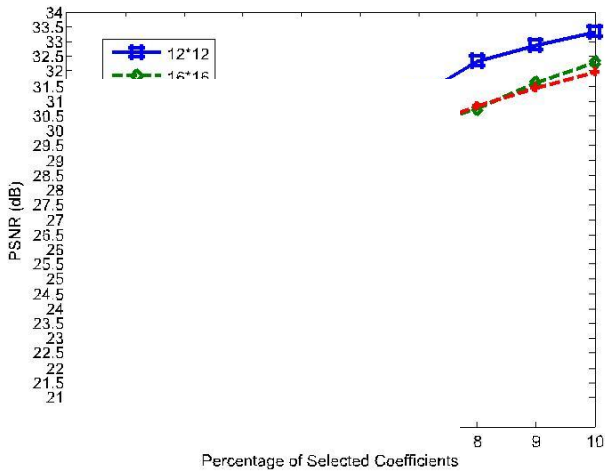


Fig. 7. Average performance of the proposed algorithms with different sizes of patches, for various percentages of selected coefficients.

chosen by experiments. The size 12 × 12, 16 × 16 and 20 × 20 are considered. There are two reasons that the size 8 × 8 is not tested. On one hand, the patches of the size are too small to contain the structure of fingerprints. On the other hand, it's difficult to compress such small patches at high compression ratios. From Fig. 7, it is seen that the size 12 × 12 works best when the number of atoms in the dictionary is 4096.

With the patches' size known, the size of the dictionary can

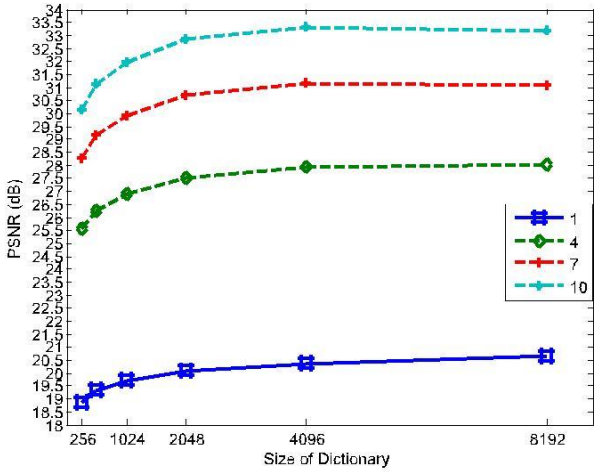


Fig. 8. The PSNR of the reconstructed images using dictionaries with different sizes at 1, 4, 7 and 10 percent of the coefficients.

be estimated by experiments. We train six dictionaries with size 144× 256, 144× 512, 144× 1024, 144× 2048, 144× 4096 and 144×8192. 1, 4, 7 and 10 percent of coefficients are tested. The results are showed in Fig. 8.

As shown in the figure, at the beginning, the quality of the reconstructed fingerprint images is improving with the size of

the dictionary increasing. This increase is sub-linear. At the point 8192, the tendency of the curves is changed. The PSNR value at point 4096 is greater than the PSNR value at the point 8192 for 7 and 10 percent of coefficients. It is because that the limited number of examples leads to over-fitting of the dictionaries.

In the following experiments, the size of patches is 12 × 12 and the size of the dictionary is 144 × 4096.

E. Comparison With Existing Fingerprint Compression Algorithms

In this section, we compare the proposed method with exist-ing fingerprint compression algorithms. Our algorithm and all others are implemented and tested on the same database. The implementation of our compression algorithm is in Matlab. We use three different image compression algorithms, JPEG, JPEG 2000 and WSQ, which have been extensively described before. The standard JPEG is a part of almost any image processing tool we do not give further reference on it. The wavelet-based JPEG 2000 we use is provided by the Matlab. The WSQ algorithm is provided by a software downloaded on the Internet for free [38].

In practice, there are fingerprint images whose sizes are not multiple of the size of the patch. If that happens, the left part of the fingerprint image is arranged into the column vectors with the same size as the patch in a given order. The order we adopt is first row and second column. These column vectors can be represented by the MP algorithm under the same dictionary.

In the experiments, it is difficult to exactly compress each fingerprint image at the given compression ratios. We choose the nearest one from those which are not less than the given

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 11, November 2015
Available at http://internationaljournalofresearch.org

TABLE II
THE MEAN VALUES AND THE VARIANCES OF THE DIFFERENCE BETWEEN THE REAL COMPRESSION RATIOS AND THE GIVEN COMPRESSION RATIOS. FOR EACH GRID, THE LEFT IS THE MEAN VALUE AND THE RIGHT IS THE VARIANCE

Compression ratio	10	15	20	25	30	35	40
DATABASE 3	1.07, 1.82	0.27, 0.04	0.52, 0.19	0.86, 0.58	0.70, 0.42	1.33, 1.83	1.71, 2.38
DATABASE 4	0.47, 0.17	0.38, 0.37	0.51, 0.13	0.68, 0.24	0.71, 0.31	0.78, 0.44	1.27, 1.87
DATABASE 5	0.30, 0.08	0.38, 0.10	0.55, 0.13	0.63, 0.29	1.22, 1.31	1.61, 2.12	3.12, 10.1

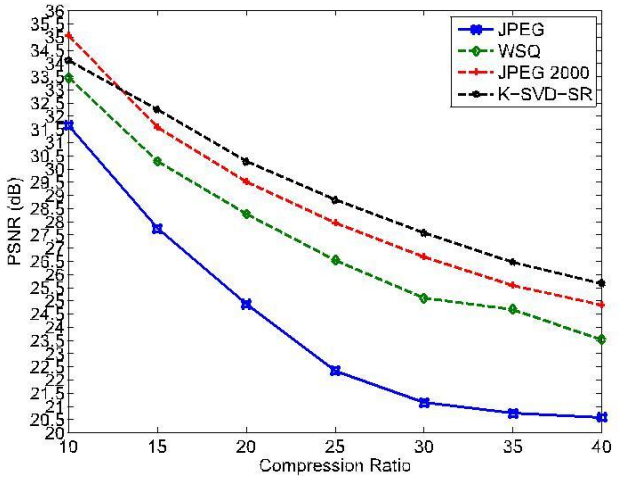


Fig. 9. Average performance of the proposed algorithms as well as JPEG, JPEG 2000 and WSQ algorithms, at various compression ratios, on DATABASE 3.

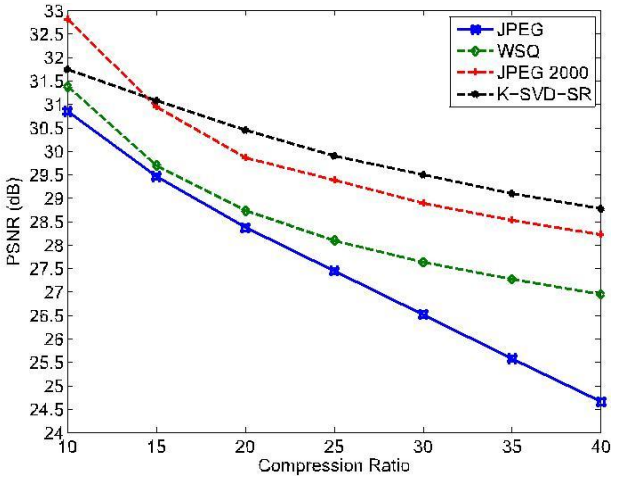


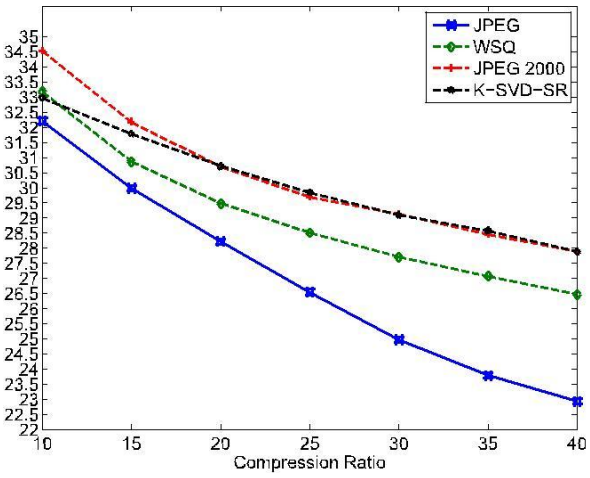
Fig. 10. Average performance of the proposed algorithms as well as JPEG, JPEG 2000 and WSQ algorithms, at various compression ratios, on DATABASE 4.

compression ratios minus 2.5 as the final compression ratio. Table II gives the mean values and the variances of the difference between the real compression ratios and the given compression ratios on DATABASE 3, DATABASE 4 and DATABASE 5.

1) Experiment Results on DATABASE 3: Fig. 9 and Table III show the average performances of our proposed algorithms, JPEG, JPEG 2000 and WSQ. In the figure, 20 on the horizontal axis means a 20 : 1 compression ratio. Compared with JPEG and WSQ, the JPEG 2000's PSNR and our method's PSNR are consistently higher. The figure shows that the proposed algorithm outperforms the JPEG 2000 algorithm when the compression ratios are high. However, at compression ratio 10 : 1, JPEG 2000 works better than ours. In our opinion, the reason is that the method based on sparse representation can't reflect the details well. This is the disadvantage of the kind of methods. When the compression ratios are high and the details are not important, these methods based on sparse representation have obvious advantage. This point also can be found in the comparative graph.

Fig. 12 demonstrates the quality of the reconstructed images obtained using different algorithms at compression ratios 30 : 1. From these reconstructed images, we can see that our algorithm achieves good visual recovery results. At the high compression ratios, the block effect of our algorithm is less serious than that of JPEG.

2) Experiment Results on DATABASE 4: Fig. 10 and Table IV show the average performances of the proposed algorithms, JPEG, JPEG 2000 and WSQ. The results on



International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 11, November 2015
Available at http://internationaljournalofresearch.org

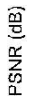


Fig. 11. Average performance of the proposed algorithms as well as JPEG, JPEG 2000 and WSQ algorithms, at various compression ratios, on DATABASE 5.

DATABASE 4 are roughly consistent with the results on DATABASE 3. Compared with JPEG and WSQ, our proposed algorithm's PSNR and JPEG 2000's PSNR are consistently higher. At compression ratio 10 : 1, JPEG2000 works better than ours, too. At compression ratio 15 : 1, the performance of our method is as good as that of JPEG 2000. At higher compression ratio, our algorithm outperforms the JPEG 2000. From the figure, we can see that the curve of our algorithm is the most flat. This means the rate of decay of our algorithm's PSNR is the slowest as the compression ratio increases.

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 11, November 2015
Available at http://internationaljournalofresearch.org

TABLE III

RESULTING AVERAGE AND VARIANCE OF PSNR (DB) FROM DIFFERENT ALGORITHMS ON DATABASE 3. FOR EACH GRID, THE LEFT IS THE MEAN VALUE AND THE RIGHT IS THE VARIANCE

Compression ratio	10	15	20	25	30	35	40
JPEG	31.65, 7.75	27.74, 6.92	24.88, 6.76	22.35, 5.91	21.15, 3.24	20.74, 1.23	20.58, 0.40
JPEG 2000	35.07, 8.48	31.59, 7.21	29.51, 6.33	27.95, 6.15	26.66, 5.76	25.60, 5.51	24.85, 5.37
WSQ	33.46, 6.47	30.23, 5.98	28.30, 8.55	26.55, 5.08	25.10, 4.57	24.69, 4.30	23.54, 4.05
K-SVD-SR	34.12, 3.24	32.25, 5.43	30.28, 5.38	28.82, 5.26	27.57, 4.85	26.47, 4.91	25.67, 4.94

TABLE IV

RESULTING AVERAGE AND VARIANCE OF PSNR (DB) FROM DIFFERENT ALGORITHMS ON DATABASE 4. FOR EACH GRID, THE LEFT IS THE MEAN VALUE AND THE RIGHT IS THE VARIANCE

Compression ratio	10	15	20	25	30	35	40
JPEG	30.85, 1.18	29.47, 1.38	28.37, 1.57	27.45, 1.72	26.52, 1.93	25.58, 2.21	24.66, 2.45
JPEG 2000	32.81, 1.24	30.95, 1.54	29.87, 1.48	29.39, 1.46	28.90, 1.37	28.54, 1.40	28.22, 1.43
WSQ	31.38, 1.30	29.70, 1.44	28.74, 1.41	28.01, 1.36	27.64, 1.31	27.27, 1.30	26.96, 1.27
K-SVD-SR	31.75, 0.74	31.08, 0.98	30.45, 1.09	29.91, 1.16	29.50, 1.17	29.10, 1.20	28.78, 1.25

(a) (b) (c) (d) (e)

Fig. 12. (a) Original image, an area at the singularities with size 300 × 300, (b) JPEG, (c) JPEG 2000, (d) WSQ, (e) the proposed method at compression ratios 30 : 1.

(a) (b) (c) (d) (e)

Fig. 13. (a) Original image with size 300 × 300, (b) JPEG, (c) JPEG 2000, (d) WSQ, (e) the proposed method at compression ratios 40 : 1.

TABLE V

RESULTING AVERAGE AND VARIANCE OF PSNR (DB) FROM DIFFERENT ALGORITHMS ON DATABASE 5. FOR EACH GRID, THE LEFT IS THE MEAN VALUE AND THE RIGHT IS THE VARIANCE

Compression ratio	10	15	20	25	30	35	40
JPEG	32.21, 4.57	29.99, 5.21	28.22, 5.67	26.54, 6.82	24.68, 7.31	23.79, 6.70	22.94, 5.15
JPEG 2000	34.54, 4.53	32.18, 4.69	30.70, 4.56	29.69, 4.52	29.13, 4.57	28.47, 4.67	27.90, 4.85
WSQ	33.18, 4.06	30.86, 4.15	29.47, 4.13	28.51, 4.21	27.71, 4.26	27.08, 4.27	26.47, 4.40
K-SVD-SR	32.97, 2.84	31.78, 3.75	30.71, 4.23	29.86, 4.50	29.30, 4.61	28.89, 4.61	28.39, 5.24

According to the trend, at higher compression ratios, our algorithm is superior to others.

Fig. 13 shows an example from the DATABASE 3 at compression ratio 40 : 1.

3) Experiment Results on DATABASE 5: Fig. 11 and Table V show the average performances of the proposed algorithms, JPEG, JPEG 2000 and WSQ. The performance of our method on this database is worse than those on DATABASE 4 and DATABASE 5. Compared with the JPEG

results, the PSNR of three other methods are consistently higher. However, at compression ratio 10 : 1, JPEG 2000 and WSQ work better than ours. At compression ratio 15 : 1, the performance of our method is less than that of JPEG 2000. At higher compression ratio, our algorithm has the same results as JPEG 2000 and outperforms WSQ. There are two reasons that the proposed method has lower performance on DATABASE 5. One is that the background of fingerprint images in DATABASE 5 is complicated. The other is the

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 11, November 2015
Available at http://internationaljournalofresearch.org

Fig. 14. (a) Original image with size 328 × 364, (b) JPEG, (C) JPEG 2000, (d) WSQ, (e) the proposed method at compression ratios 40 : 1.

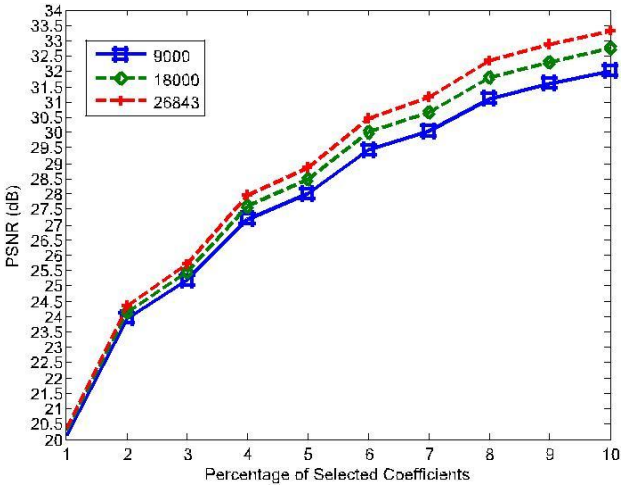


Fig. 15. The effect of the number of the training samples on average performance of the K-SVD algorithm, for various percentages of selected coefficients. Three numbers of the training samples, 9000, 18000 and 26843, are considered.

quality of these fingerprint images is relatively poor. It is difficult to represent a bad patch well.

Fig. 14 shows an example from DATABASE 5 at compression ratio 40 : 1.

F. How to Design the Training Set

From the above experiments, we can see that the dictionary is very important. Suppose that the size of the dictionary and the patches are $M \times N$ and $m \times n$ respectively. There are two parameters about the dictionary, namely, M and N. In fact, $M = m \times n$ is a variable related to size of the patches. N is the number of the feature set (In the first two ways to construct the dictionary, N is the number of the selected patches. In the third way, N is the number of the feature trained by the K-SVD method). In theory, the feature set should be large enough that the patches can be represented sparsely, that is to say, N should be sufficiently large. The bigger M is, the higher the compression efficiency is. However, as M increases, for good compression effect, N also does. Large N means the high computational complexity. After all, these two issues, sufficiency and efficiency, are complementary. The method of estimating these parameters are given before. In addition, targeted at the problem of fingerprint, there are several other details to be considered.

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 11, November 2015
Available at http://internationaljournalofresearch.org

1) The Number of Training Samples: For the third way in which the dictionary is trained by the K-SVD method, we consider the effect of the number of the training samples on the PSNR. Here, the test set is DATABASE 2 and the size of the dictionaries is 144 × 4096. From Fig. 15, we can see that the higher PSNR are, the larger the number of the training samples is. To obtain a good dictionary, numerous training samples are necessary. When the number of the train-ing samples is about 27000, the performance of the proposed algorithm is good enough. The figure also reflects that the proposed method has the ability of learning by increasing the number of training samples.

2) Components of the Training Set: General images are too complicated to deal with. Fingerprint images are relatively simple. Nevertheless, there are many situations to be consid-ered. In the local, orientation, ridge frequency and minutiae are the main feature of the fingerprints. These factors should be taken into consideration when we construct the dictionary. Fig. 16 shows these patches.

3) How to Modify the Dictionary: As we mentioned before, our algorithm can make use of the empirical information. In other words, it has the ability of learning. After a period of application, there will be many patches which can't be represented well. We can use these new patches and original patches to update the dictionary. Suppose that the original dictionary is D_0 and the original training set is Y_0. The set of new patches which can't be represented well is Y_1. We put the the original training set and the new set together to produce a larger training set (referred as $Y = [Y_0, Y_1]$). We can solve the following optimization problem to obtain the new dictionary. In the update, we take the old dictionary as the initial matrix of the dictionary, namely, D_0.

$$\min_{D,X} \|Y - DX\|_F^2 \quad s.t. \ \forall i, \|X_i\|_0 < T \tag{9}$$

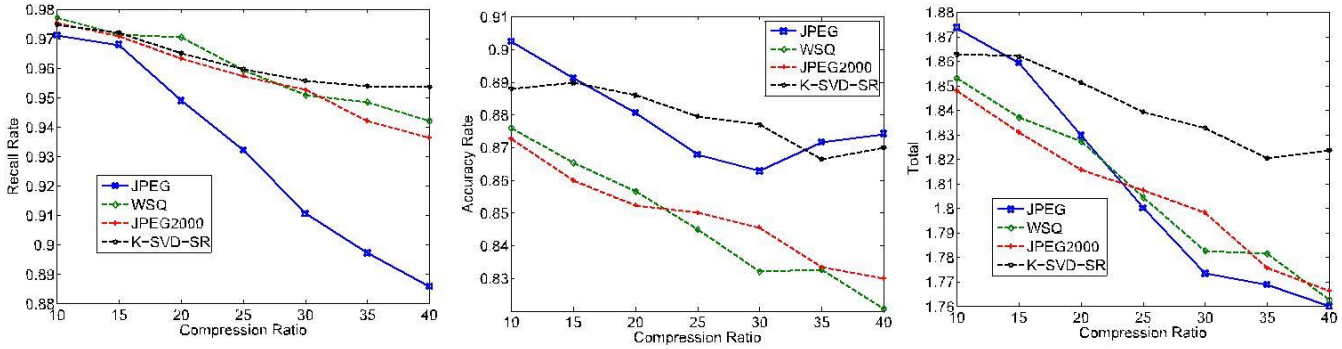
G. Robustness

Though there are successful approaches [39] based on Gabor filtering, which don't rely on the minutiae. However, in most Automatic Fingerprint Identification System (AFIS), the main feature used to match two fingerprint images is minutiae. Therefore, we illustrate the robustness of our algorithm by comparing the difference of minutiae between pre- and post-compression. There are three important standards to measure the robustness, namely, the accuracy rate (AR), the recall rate (RR) and their sum.

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 11, November 2015
Available at http://internationaljournalofresearch.org

(a) (b) (c)

Fig. 16. (a) 64 patches with different orientations, (b) 64 patches with different ridge frequency, (c) 64 unusual patches which show the minutiae, the edge of the fingerprint and others.



(a) (b) (c)

Fig. 17. (a) The Recall Rate, (b) The Accuracy Rate, (c) the Total Sum of the Recall Rate and the Accuracy Rate, for different algorithms at various compression ratios.

A minutiae is a triple (x, y, ϑ), where x, y, and ϑ represent its horizontal coordinate, vertical coordinate and orientation, respectively. We say a minutiae (x_1, y_1, ϑ_1) exists in a fingerprint if there is a minutiae (x_2, y_2, ϑ_1) in the fingerprint that satisfies the following conditions: $(x_1 - x_2)^2 + (y_1 - y_2)^2 < \alpha$ and $|\vartheta_1 - \vartheta_2| < \beta$, where α and β are thresholds.

Suppose that there are *Num* minutiae in the original image, Num^- minutiae in the reconstruction and Num_1 minutiae in the reconstruction also exist in the original image.

The accuracy rate: $AR = \frac{Num_1}{Num^-}$

The recall rate: $RR = \frac{Num_1}{Num}$

The total rate: $Total = AR + RR$

We test the robustness in DATABASE 3. Results can be seen in Fig. 17.

From Fig. 17, we can see that our algorithm achieves 87% in precision and 95% in recall even at compression ratio 40 : 1. Fig. 17 shows that the Recall Rate of our algorithm is high. From Fig. 17, it can be seen that the Accuracy Rates of the JPEG algorithm and our algorithm are consistently superior to those of other algorithms. Meanwhile, Fig. 17 demonstrates that the Total Rate of our algorithm is the best except at compression ratio 10 : 1. The Accuracy Rate indicates that our algorithm can reduce the spurious minutiae points while the Recall Rate shows that our algorithm can hold most of the minutiae robustly during the compression and reconstruction.

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 11, November 2015
Available at http://internationaljournalofresearch.org

From the above experiments, we find that the JPEG algorithm has a bad performance measured in terms of PSNR. However, from the point of the Recall Rate and the Accu-racy Rate, the algorithm has relatively good performance. This suggests that only PSNR is not enough to measure the performance of compression algorithms. At least, it is true for fingerprint compression. According to different purposes and different kinds of images, more evaluation indexes are necessary, such as the Accuracy Rate, the Recall Rate for fingerprint image.

VI. Conclusion

A new compression algorithm adapted to fingerprint images is introduced. Despite the simplicity of our 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 experiments show that the block effect of our algorithm is less serious than that of JPEG. We consider the effect of three different dictionaries on fingerprint compression. The experiments reflect that the dictionary obtained by the K-SVD algorithm works best. Moreover, the larger the number of the training set is, the better the compression result is.

One of the main difficulties in developing compression algorithms for fingerprints resides in the need for preserving the

minutiae which are used in the identification. The experiments show that our algorithm can hold most of the minutiae robustly during the compression and reconstruction.

There are many intriguing questions that future work should consider. First, the features and the methods for constructing dictionaries should be thought over. Secondly, the training samples should include fingerprints with different quality ("good", "bad", "ugly"). Thirdly, the optimization algorithms for solving the sparse representation need to be investigated. Fourthly, optimize the code to reduce complexity of our proposed method. Finally, other applications based on sparse representation for fingerprint images should be explored.

ACKNOWLEDGMENT

The authors would like to thank the Associate Editor and the anonymous reviewers for their valuable comments and suggestions that helped greatly improve the quality of this paper.

REFERENCES

[1] D. Maltoni, D. Miao, A. K. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*, 2nd ed. London, U.K.: Springer-Verlag, 2009.
[2] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. C-23, no. 1, pp. 90–93, Jan. 1974.
[3] 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.
[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] 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.
[7] 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.
[8] 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.
[9] 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.
[10] 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.
[11] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.
[12] S. S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM Rev.*, vol. 43, no. 1, pp. 129–159, 2001.
[13] 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.
[14] 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.
[15] S. Agarwal and D. Roth, "Learning a sparse representation for object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2002, pp. 113–127.
[16] 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.
[17] K. Skretting and K. Engan, "Image compression using learned dictionar-ies by RLS-DLA and compared with K-SVD," in *Proc. IEEE ICASSP*, May 2011, pp. 1517–1520.
[18] 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.
[19] 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.
[20] 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.
[21] A. J. Ferreira and M. A. T. Figueiredo, "On the use of independent component analysis for image compression," *Signal Process., Image Commun.*, 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 non-negative matrix factorization," *Nature*, vol. 401, pp. 799–791, Oct. 1999.
[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] 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.
[27] E. Candes, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Commun. Pure Appl. Math.*, vol. 59, no. 8, pp. 1207–1223, 2006.
[28] E. Candes, J. Romberg, and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.
[29] D. Donoho, "For most large underdetermined systems of linear equations the minimal _1-norm solution is also the sparsest solution," *Commun. Pure Appl. Math.*, vol. 59, no. 6, pp. 797–829, 2006.
[30] E. Candes and Y. Plan, "Near-ideal model selection by _1 minimization," *Ann. Statist.*, vol. 37, no. 5A, pp. 2145–2177, 2009.
[31] I. F. Gorodnitsky and B. D. Rao, "Sparse signal reconstruction from lim-ited data using FOCUSS: A re-weighted norm minimization algorithm," *IEEE Trans. Signal Process.*, vol. 45, no. 3, pp. 600–616, Mar. 1997.
[32] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, 2009.
[33] M. Aharon, M. Elad, and A. M. Bruckstein, "On the uniqueness of overcomplete dictionaries, and a practical way to retrieve them," *J. Linear Algebra Appl.*, vol. 416, no. 1, pp. 48–67, 2006.
[34] 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.
[35] K. Sayood, *Introduction to Data Compression*, 3rd ed. San Mateo, CA, USA: Morgan Kaufman, 2005, pp. 81–115.
[36] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982.
[37] (2013, Jun.). *Fingerprint Images* [Online]. Available: http://pan.baidu.com/share/link?shareid= 3872475108&uk=2301599420
[38] (2012, Dec.). *SetupWSQ* [Online]. Available: http://www.download3k.com/Install-WSQ-viewer.html
[39] S. Prabhakar, "Fingerprint classification and matching using a filterbank," Ph.D. dissertation, Dept. Comput. Sci., Eng., Michigan State Univ., East Lansing, MI, USA, 2001.