

# An AK-BRP dictionary learning algorithm for video frame sparse representation in compressed sensing

Yang Qian<sup>1</sup>  · Lei Li<sup>2</sup> · Zhenzhen Yang<sup>2</sup> · Feifei Zhou<sup>1</sup>

Received: 17 January 2016 / Revised: 9 September 2016 / Accepted: 7 November 2016 /

Published online: 22 November 2016

© Springer Science+Business Media New York 2016

**Abstract** Sparsifying transform is an important prerequisite in compressed sensing. And it is practically significant to research the fast and efficient signal sparse representation methods. In this paper, we propose an adaptive K-BRP (AK-BRP) dictionary learning algorithm. The bilateral random projection (BRP), a method of low rank approximation, is used to update the dictionary atoms. Furthermore, in the sparse coding stage, an adaptive sparsity constraint is utilized to obtain sparse representation coefficient and helps to improve the efficiency of the dictionary update stage further. Finally, for video frame sparse representation, our adaptive dictionary learning algorithm achieves better performance than K-SVD dictionary learning algorithm in terms of computation cost. And our method produces smaller reconstruction error as well.

**Keywords** Bilateral random projections (BRP) · Adaptive K-BRP algorithm · Dictionary learning · Sparse representation · K-SVD algorithm

## 1 Introduction

Recently, solving the inverse problem of images has attracted extensive attention of scholars [4, 9]. In the context of imaging, several methods have been proposed to exploit the sparsity of image patches in a sparsifying transform domain or learned dictionary to reconstruct images, especially in the field of Compressed Sensing (CS) [11]. CS conducts sampling and compres-

---

✉ Yang Qian  
739086359@qq.com

<sup>1</sup> School of Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China

<sup>2</sup> Center for Visual Cognitive Computation and Its Application, Nanjing University of Posts and Telecommunications, Nanjing 210023, China

sion at the same time by utilizing the redundancy of the signal. This technology mainly consists of three important stages, i.e., signal sparse representation, sensing matrix design and signal reconstruction. And finding the best sparsifying basis is the prerequisite for compressed sensing theory.

The conventional CS sparse representation methods often utilize fixed sparsifying transforms such as Fourier transform, discrete cosine transform, wavelet transform, etc. However, for lacking of translation and rotation invariance, the orthogonal basis is not enough to capture the various features of images. In recent years, with the in-depth study of dictionary learning algorithms [8, 20, 21], researchers have extended CS to the redundant dictionary, which is adaptively learned from the processed signal itself. Hence, the sparse decomposition of signals in redundant dictionaries has become a research hotspot in sparse coding field. Various studies have shown that the learned dictionaries can commendably match the structure of the signal or image itself, and exhibit great performance in image classification [3], image denoising [13, 17], image super-resolution [15], etc.

The K-singular value decomposition (K-SVD) [1] algorithm, an iterative method that alternates between sparse coding stage and dictionary atoms update stage, is one of the most well-known dictionary learning algorithms. It has been widely used in face recognition [14, 22, 25], image processing [2, 18] and so on. This method updates the dictionary columns jointly with an update of the sparse representation coefficients related to it. Thus it helps to avoid matrix inversion calculation and accelerate convergence. However, the K-SVD algorithm needs to calculate SVD excessively for the whole dictionary atoms during each iteration procedure. Since SVD operation has large computational cost as the dimensions of the input vector increases, the speed of the K-SVD algorithm may slow down. Furthermore, only the largest singular value and its corresponding singular vector are used while the others are abandoned directly, which results in the waste of computing resources.

Ref. [26] proposed the K-RBP dictionary learning algorithm, a novel dictionary update method, to sparsely represent video frame. This algorithm makes changes of K-SVD in the dictionary update stage and improves the manner of calculating the largest singular value and its corresponding singular vector. Based on Ref. [26], we exploit a novel dictionary learning algorithm, called adaptive K-bilateral random projections (AK-BRP), to relieve the above limitations of the K-SVD algorithm. In the sparse coding stage, the sparsity constraint is associated with the current dictionary to obtain an adaptive sparsity constraint; while in the dictionary update stage, the bilateral random projections (BRP) [6, 7], a method of low rank approximation, is used to directly compute the largest singular value and its corresponding singular vector. Besides, the adaptive sparsity is employed by associating the maximal number of dictionary atoms in the sparse coding stage with the iterated updated dictionaries. Using the dictionaries learned by our new method as sparse representation for video frame compressed sensing, experimental results on a wide range of video frames for CS recovery have shown that our proposed algorithm is better than K-SVD algorithm in terms of computation expense, running time and sparse representation.

The remainder of this paper is organized as follows. Section 2 briefly reviews CS theory and the K-SVD dictionary learning algorithm. Section 3 provides the proposed dictionary learning algorithm in detail. The compressed sensing algorithm based on our adaptive redundant dictionary learning for video frame recovery is detailed in Section 4. Experimental results are reported in Section 5. Finally, in Section 6, we conclude this paper.

## 2 Background

### 2.1 Compressed sensing

Compressed sensing theory indicates that if a signal  $x \in R^{n \times 1}$  is sparse or compressible in an orthogonal basis or tight framework  $\Psi = (\psi_1, \dots, \psi_n)^T$ , (here  $\Psi$  satisfies  $\Psi \cdot \Psi^T = I$ ), we can use a random sensing matrix  $\Phi \in R^{m \times n}$  ( $m < n$ ), which is irrelevant to the sparsifying transform base, to obtain the linear observational vector  $y \in R^{m \times 1}$  of the sparse transform coefficient vector  $\Theta = \Psi^T x$ . Then the original signal  $x$  can be exactly reconstructed from the linear measurements  $y$ , whose number is much smaller than that of the original signal. The CS model can be formulated as

$$y = \Phi x = \Phi \Psi \Theta \quad (1)$$

It has been well known that the sparsity degree of the signal, which is a key to achieve accurate reconstruction, plays an important role in the field of compressed sensing. The higher degree of a signal, the higher reconstruction precision it will have. Thus, finding a domain in which the signal has a high degree of sparsity is one of the main challenges CS recovery should face [24]. The conventional CS recovery algorithms mainly use a set of fixed sparsifying domains such as DCT, wavelet and FFT domain. These algorithms are not able to capture the various geometrical features contained in the high dimensional data and result in poor CS recovery performance for high dimensional signals. Recently, with the rapid development of signal sparse representation methods based on over-complete dictionaries, the study of CS sparse coding is mostly extended to redundant dictionaries. Nowadays, designing a more efficient redundant dictionary has become the emphasis.

Another fundamental principle that CS relies on is incoherent projection [5]. CS theory requires that  $\Phi$ , the sensing matrix, and  $\Psi$ , the sparse basis, must satisfy Restricted Isometry Property (RIP) [4, 5]. Only in this way can we reconstruct the original signal accurately.

The purpose of CS reconstruction is to recover  $\Theta$  from the linear measurements  $y$ , and then obtain the approximation of the original signal by  $x = \Psi \Theta$ . It can be described as the following optimization problem

$$\min_{\Theta} \|y - \Phi \Psi \Theta\|_2^2 + \lambda \|\Theta\|_0 \quad (2)$$

where  $\lambda$  is a non-negative parameter,  $\|v\|_0$  is the  $l_0$  norm, counting the number of nonzero elements of the vector  $v$ , and  $\|v\|_2$  is the  $l_2$  norm which is defined as  $\|v\|_2 = \sqrt{\sum v_i^2}$ .

Though Eq. (2) is a NP-hard problem, greedy algorithms, such as orthogonal matching pursuit (OMP), can obtain approximate solution with relatively small running time. When  $l_0$  norm is replaced with  $l_1$  norm, which adds all the absolute values of the entries in the vector, the minimization problem can be solved by the convex relaxation methods such as BP, least absolute shrinkage method and LASSO.

### 2.2 Dictionary learning

In this sub-section, we will describe the K-SVD algorithm, which is one of the most commonly used methods of dictionary learning, in detail. Given a training set  $X \in R^{n \times N}$  whose columns are the form of  $\{x_i\}_{i=1}^N$ , and  $A = \{a_i\}_{i=1}^N \in R^{K \times N}$  is the corresponding coefficient. Then, the

dictionary learning process is to find a possible optimal dictionary  $D \in R^{n \times K} (K > n)$ , which contains  $K$  prototype signal atoms for columns  $\{d_j\}_{j=1}^K$ , for sparse representation of the given training samples  $X$  It can be expressed as

$$\min_{D,A} \|X - DA\|_F^2 \text{ s.t. } \forall i, \|a_i\|_0 \leq T_0 \tag{3}$$

where  $\|\cdot\|_F$  denotes the Frobenius norm of a matrix,  $\|\cdot\|_0$  is the  $l_0$  norm, and  $T_0$  is the maximum number of non-zero atoms used in the sparse representation vector.

The specific implementation steps of the K-SVD algorithm is outlined in Algorithm 1 [1]:

---

**Algorithm 1: K-SVD**

---

**Input:** training samples  $X = \{x_i\}_{i=1}^N$ , max number of iterations  $P$  and sparsity threshold  $T_0$

**Initialization:** set the dictionary matrix  $D^{(0)} \in R^{n \times K}$  with  $l_2$  normalized columns

For  $j = 1, 2, \dots, P$

A. Sparse coding stage

Use any pursuit algorithm to compute the representation vectors  $a_i$  under the fixed dictionary  $D$ , by solving the following constrained optimization

$$\min_{a_i} \|x_i - Da_i\|_2^2 \text{ s.t. } \|a_i\|_0 \leq T_0, \forall i = 1, 2, \dots, N \tag{4}$$

B. Dictionary update stage

Fix the sparse representation matrix  $A$ , for each column  $k = 1, 2, \dots, K$

- 1) Define the index set of examples that use atom  $d_k$  :  
 $\omega_k = \{i \mid 1 \leq i \leq N, a_i(k) \neq 0\}$ ;
- 2) Compute  $E_k$  by  $E_k = X - \sum_{j \neq k} d_j a_j = X - (DA - d_k a_k)$ , wherein  $E_k$  means the overall representation error matrix;
- 3) Constraint  $E_k$  corresponding to  $\omega_k$ , and then obtain  $E_k^R$  by  $E_k^R = E_k \Omega_k$ , wherein  $\Omega_k$  is a matrix of size  $N \times |\omega_k|$  with ones on the  $(\omega_k(i), i)$ th entries and zeros elsewhere;
- 4) Decompose  $E_k^R$  with singular value decomposition (SVD) by  $E_k^R = U \Delta V^T$  ;  
 Update the dictionary column and the coefficient vector respectively:

$$d_{k,updated} = u_1$$

$$a_{k,updated} = \Delta(1,1) \cdot v_1$$

**Output:** the trained dictionary  $D = D_p$

---

### 3 The proposed algorithm

From the process of the K-SVD algorithm, it is not difficult to find that the singular value decomposition (SVD), which is used for low rank approximation, plays a significant role in K-SVD. However, the use of SVD restrains K-SVD in terms of execution time. Especially when the dimension of the input vectors increases, the computation time will rapidly increase, which worsen the K-SVD algorithm. Consequently, we would like to propose a novel fast dictionary

learning algorithm named adaptive K-BRP algorithm. Since dictionary learning algorithm is usually treated as iterations of sparse coding stage and dictionary update stage, we will describe our new dictionary learning algorithm in these two stages.

### 3.1 Sparse coding stage

Unlike the sparse coding stage of K-SVD algorithm, in our method, we associate the sparsity upper-bound with the iterated updated dictionary by using the coherence of the current dictionary, to obtain an adaptive sparsity constraint, sequentially reducing the reconstruction errors iteratively.

Suppose that  $T_j$  is the sparsity upper-bound for each iteration process,  $T_j$  denotes

$$T_j = \frac{1}{2} \left( 1 + \frac{1}{\mu(D_j)} \right) \quad j = 1, 2, \dots, P \quad (5)$$

where  $\mu(D)$  is the mutual coherence of the dictionary  $D$ , which describes the maximum similarity between the over-complete dictionary atoms, expressed as [10]

$$\mu(D) = \max_{i \neq j} |\langle d_i, d_j \rangle| \quad (6)$$

Here, the  $\mu(D)$  function is valued between 0 and 1. The minimum is reached for an orthogonal dictionary and the maximum for a dictionary containing at least two collinear atoms [16].

In order to expound the rationality of our adaptive sparsity constraint  $T_j$ , the following theorem is introduced [10, 12]:

**Theorem 3.1** Let  $D \in \mathbb{R}^{n \times K}$  ( $K > n$ ) be the dictionary and  $\mu(D)$  denote the mutual coherence of the dictionary  $D$ . Suppose that  $x = Da$ , where  $a$  is sparse, if the sparsity  $S$ , the number of nonzero entries of the correct coefficients, satisfies Eq. (7)

$$S < \frac{1}{2} \left( 1 + \frac{1}{\mu(D)} \right) \quad (7)$$

then the following conclusions can be deduced:

- 1) The solution  $a$  of the minimization problem (4) must be the sparsest and unique;
- 2) The  $l_0$  norm minimization problem in Eq. (3) can be equivalent to the  $l_1$  norm;
- 3) Any pursuit algorithm (such as OMP) can find out the best linear combination of  $S$  atoms from the dictionary  $D$

Hence, the defined  $T_j$  can ensure precise reconstruction of the sparse signal, and is reasonably practicable.

Replacing  $T_0$  in Eq. (4) with  $T_j$ , then the sparse coding stage is to solve the following optimization problem

$$\min_{a_i} \|x_i - D_j a_i\|_2^2 \quad s.t. \quad \|a_i\|_0 \leq T_j \quad \forall i = 1, 2, \dots, N; \quad j = 1, 2, \dots, P \quad (8)$$

This problem can be solved by greedy algorithms, such as OMP, ROMP, etc.

### 3.2 Dictionary update stage

As is known to all, K-SVD algorithm employs singular value decomposition (SVD), a low rank approximation method, to obtain the update of dictionary atoms, since SVD can achieve the minimum reconstruction error. However, the use of SVD restricts K-SVD algorithm in terms of computation time, especially for high dimensional data. To this end, we would like to adopt another method of low rank approximation, named bilateral random projections (BRP) [6, 7], to reduce the computation time.

Given  $E_k^R \in R^{n \times |\omega_k|}$ , the restricted representation error matrix that corresponds to examples that use the atom  $d_k$  in the update stage of the K-SVD algorithm, then the  $r$  rank bilateral random projections (BRP) of  $E_k^R$  is described as

$$\begin{cases} Y_1 = E_k^R A_1 \\ Y_2 = (E_k^R)^T A_2 \end{cases} \tag{9}$$

wherein  $A_1 \in R^{|\omega_k| \times r}$  and  $A_2 \in R^{n \times r}$  are random matrices.

Then the fast rank- $r$  approximation of  $E_k^R$  can be expressed as

$$\hat{E}_k^R = Y_1 (A_2^T Y_1)^{-1} Y_2^T \tag{10}$$

However, it has been pointed out in reference [6] that when singular values of  $E_k^R$  decay slowly, Eq. (10) may perform poorly. At this point, we need to apply power scheme [19] to  $\hat{E}_k^R$  for improving the approximation precision. In the power scheme modification, we instead calculate the BRP of a matrix  $\tilde{E}_k^R = (E_k^R (E_k^R)^T)^q E_k^R$ , whose singular values decay faster than  $E_k^R$ . Particularly,  $\lambda_i(\tilde{E}_k^R) = \lambda_i(E_k^R)^{2q+1}$ , here  $\lambda_i(\cdot)$  denotes the  $i$ th singular value of the matrix. Both  $E_k^R$  and  $\tilde{E}_k^R$  share the same singular vectors. Thereafter, we will be able to obtain the BRP of  $\tilde{E}_k^R$  according to that of  $E_k^R$  as

$$\begin{cases} Y_1 = \tilde{E}_k^R A_1 \\ Y_2 = (\tilde{E}_k^R)^T A_2 \end{cases} \tag{11}$$

According to Eq. (10), the BRP based  $r$  rank approximation of  $\tilde{E}_k^R$  is

$$\hat{\tilde{E}}_k^R = Y_1 (A_2^T Y_1)^{-1} Y_2^T \tag{12}$$

In order to obtain the fast  $r$  rank approximation of  $E_k^R$ , we need to calculate the QR decomposition of  $Y_1$  and  $Y_2$  respectively

$$\begin{cases} Y_1 = Q_1 R_1 \\ Y_2 = Q_2 R_2 \end{cases} \tag{13}$$

The fast low-rank approximation of  $E_k^R$  with rank  $r$  is then given by

$$\hat{E}_r = \left( \hat{\tilde{E}}_k^R \right)^{\frac{1}{2q+1}} = Q_1 \left[ R_1 (A_2^T Y_1)^{-1} R_2^T \right]^{\frac{1}{2q+1}} Q_2^T \tag{14}$$

Because in each dictionary atom update process, only the largest singular value and its corresponding singular vector are used, we just need to calculate the fast 1-rank approximation of  $E_k^R$ . Then, the random matrices  $A_1$  and  $A_2$  can be transformed into random vectors, which will greatly accelerate the speed of the algorithm.

#### 4 Compressed sensing algorithm based on the AK-BRP dictionary learning

In this section, we will elaborate the scheme of video frame compressed sensing based on the proposed dictionary learning algorithm in detail.

Assuming that a video sequence is composed of  $I$  frames, of size  $W \times L$ , then the video sequence can be expressed as

$$squ = f_r(x, y) \quad (15)$$

where  $1 \leq x \leq L, 1 \leq y \leq W, 1 \leq r \leq I$ ,

Dividing each frame of the video sequence into small non-overlapping blocks of equal size (i.e., size  $\sqrt{n} \times \sqrt{n}$ ), here we suppose that  $W$  and  $L$  are both the integral multiple of  $\sqrt{n}$ , then each frame can be divided into  $WL/n$  blocks. Among them, the corresponding vector  $x_r^t$  of the  $t$ th image patch  $f_r^t(x, y)$  in the  $r$ th frame can then be formed by the way of row stacking or column stacking, expressed as

$$x_r^t = (f_r^t(1, 1), f_r^t(1, 2) \cdots f_r^t(1, \sqrt{n}), f_r^t(2, 1), \cdots f_r^t(2, \sqrt{n}) \cdots f_r^t(\sqrt{n}, 1) \cdots f_r^t(\sqrt{n}, \sqrt{n}))^T \quad (16)$$

where the vector  $x_r^t \in R^{n \times 1}, 1 \leq t \leq WL/n, 1 \leq r \leq I$ ,

For each image patch, the corresponding measurements vector of CS is  $y_r^t = \Phi x_r^t + \varepsilon_r^t$ , where  $\varepsilon_r^t$  is the random noise in measurement process (here in this paper, we assume the same sampling matrix is applied to each block). In the compressed sensing process, we exploit the proposed AK-BRP dictionary learning algorithm to train the redundant dictionary  $D = D_P$  and then represent each patch  $x_r^t$  by a linear combination of  $D a_r^t$  (i.e.  $x_r^t = D a_r^t$ ), wherein  $D \in R^{n \times K}$  (each atom of  $D$  has unit norm),  $a_r^t \in R^{K \times 1}$  is the sparse representation coefficient. Because the sparsest representation of  $x_r^t$  corresponds to the vector of weight coefficient  $a_r^t$  with the smallest number of non-zeros, we use greedy iterative algorithms, such as OMP, to find the coefficient vector  $a_r^t$ , and then reconstruct the  $t$ th block by  $x_r^t = D a_r^t$ . And lastly, the reconstructed  $r$ th frame of the video sequence can be obtained by reshaped every  $t$ th block in the  $r$ th frame. The whole procedure is as follows

#### 5 Experiment simulations and analysis

In this section, we carry out several experimental simulations on two standard CIF (frame size:  $352 \times 288$ , luminance only) video sequences: Foreman and Akiyo, to evaluate the performance of our proposed dictionary learning algorithm for CS sparse representation.

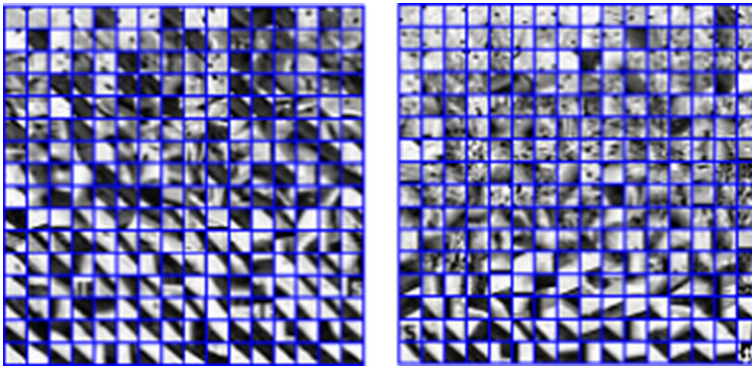
**Algorithm 2: AK-BRP-CS**

- 
- (1) **Input:** training samples  $\{x_i\}_{i=1}^N$ , max number of iterations  $P$ , measure matrix  $\Phi$  and linear measurement  $y_r^t$  of the  $t$ th block in the  $r$ th frame
- (2) **Output:** the trained dictionary  $D = D_p$  and reconstructed  $r$ th video frame  $\hat{X}_r$
- (3) **Initialization:** set the dictionary matrix  $D^{(0)} \in \mathbb{R}^{n \times K}$  with  $l_2$  normalized columns and obtain the initial sparsity  $T_0$  using Equation (5)
- (4) For  $j = 1, 2, \dots, P$
- A. Sparse coding stage  
Use OMP algorithm to compute the sparse coefficient  $A_j$  under the fixed current dictionary  $D_{j-1}$  and sparsity constraint  $T_{j-1}$ , by solving Equation (8)
  - B. Dictionary update stage  
Given  $A_j$ , for each column  $k = 1, 2, \dots, K$  in  $D_{j-1}$ 
    - a. Define the index set of examples that use atom  $d_k$ :  $\omega_k = \{i \mid 1 \leq i \leq N, a_i(k) \neq 0\}$ ;
    - b. Compute overall representation error matrix  $E_k = X - \sum_{j \neq k} d_j a_j$ ;
    - c. Restrict  $E_k$  by choosing only the columns corresponding to  $\omega_k$ , and then obtain  $E_k^R$  by  $E_k^R = E_k \Omega_k$ , wherein  $\Omega_k$  is a matrix of size  $N \times |\omega_k|$  with ones on the  $(\omega_k(i), i)$ th entries and zeros elsewhere;
    - d. Apply BRP algorithm to get  $E_k^R$ th 1-rank approximation  $\hat{E}_r$ ;
    - e. Update the dictionary column and the coefficient vector respectively by:
 
$$d_{k,updated} = \hat{E}_r(:, 1) / \text{norm}(\hat{E}_r(:, 1))$$

$$a_{k,updated} = \hat{E}_r(1, :) \cdot \text{norm}(\hat{E}_r(1, :))$$
- Obtain the updated dictionary  $D_j$   
Compute the sparsity  $T_j$  by Equation (5)  
End
- (5) Given  $\Phi$ ,  $y_r^t$ , compute coefficient  $\hat{a}_r^t$  in sparsifying transform  $D = D_p$ , by Equation (1)
- (6) Obtain the  $t$ th reconstructed image patches, using  $x_r^t = D a_r^t$
- (7) Rebuild the reconstructed  $r$ th frame  $\hat{X}_r$  from  $\hat{x}_r^t$
- 

In our experiments, each test frame is divided into  $8 \times 8$  non-overlapping blocks and the CS measurements are obtained by applying a Gaussian random projection matrix to each block (here, each block is sampled by the same sensing matrix). The other parameter setting of proposed algorithm is as follows: the over-complete dictionary size is chosen to be  $64 \times 256$  and the number of iterations for training is set to 30. All implementations involved in the experiments are coded in Matlab R2011a. Computations are performed on a ThinkPad





**Fig. 1** Visual comparison of two types of learned dictionaries. From left to right: the redundant dictionary trained by K-SVD algorithm; the redundant dictionary trained by AK-BRP algorithm

computer with Intel (R) Core (TM) i5-5250U CPU @ 1.6 GHz, 8GB memory, and Windows 8 operating system.

### 5.1 Visual effect of dictionary learning

In order to validate that the improved dictionary learning algorithm has better performance in sparse representation, the K-SVD dictionary learning algorithm and the proposed AK-BRP dictionary learning algorithm are respectively used to represent the 23th frame of the Foreman video sequence by learning dictionaries from the processed frame itself. In the experiment, the sparsity of K-SVD is set to be 5, where sparsity is defined as the maximum number of non-zero coefficients used to represent each block in the sparse coding stage. The redundant dictionaries trained by these two algorithms are shown in Fig. 1.

### 5.2 Improvement on accuracy in video frame compressed sensing

In this sub-section, we will show the influence of the proposed dictionary learning algorithm on the performance of CS recovery. In our simulation, the K-SVD redundant dictionary and the AK-BRP redundant dictionary are respectively selected as the sparsifying basis of CS, and the OMP algorithm is used as CS reconstruction algorithm to restore every test frame. For conveniently describing, the two methods are credited as KSVD-CS and AK-BRP-CS, respectively.

**Table 1** Time Comparison with Various CS Methods Based on Different Sparsifying Basis

Ratio	Algorithms	foreman1	foreman6	foreman10	foreman15	foreman23	Average
r = 0.4	<b>KSVD-CS</b>	60.41634	59.49138	60.0947	59.20356	59.6849	59.77818
	<b>Proposed</b>	25.4466	25.58294	24.46592	24.22988	26.04726	<b>25.15452</b>
r = 0.5	<b>KSVD-CS</b>	56.45354	54.72946	56.02416	57.04094	56.44854	56.13933
	<b>Proposed</b>	22.27688	22.0898	22.15242	22.62102	22.00844	<b>22.22971</b>
r = 0.6	<b>KSVD-CS</b>	60.1871	59.42668	61.48126	59.55592	61.33816	60.39782
	<b>Proposed</b>	27.46088	27.3018	28.00678	27.49906	26.86626	<b>27.42696</b>

**Table 2** PSNR Comparison with Various CS Methods Based on Different Sparsifying Basis

Ratio	Algorithms	foreman1	foreman6	foreman10	foreman15	foreman23	Average
r = 0.4	<b>KSVD-CS</b>	20.60804	20.8893	21.63212	19.50334	18.6784	20.26224
	<b>Proposed</b>	25.81312	25.6872	25.7925	24.9706	25.31566	<b>25.51582</b>
r = 0.5	<b>KSVD-CS</b>	23.6426	23.94142	24.69418	21.73652	21.54268	23.11148
	<b>Proposed</b>	27.9898	27.9002	27.4323	27.22904	27.36548	<b>27.58336</b>
r = 0.6	<b>KSVD-CS</b>	24.33136	25.55038	25.94274	23.99478	23.9586	24.75557
	<b>Proposed</b>	29.00098	28.4917	28.49266	27.77302	28.0987	<b>28.37141</b>

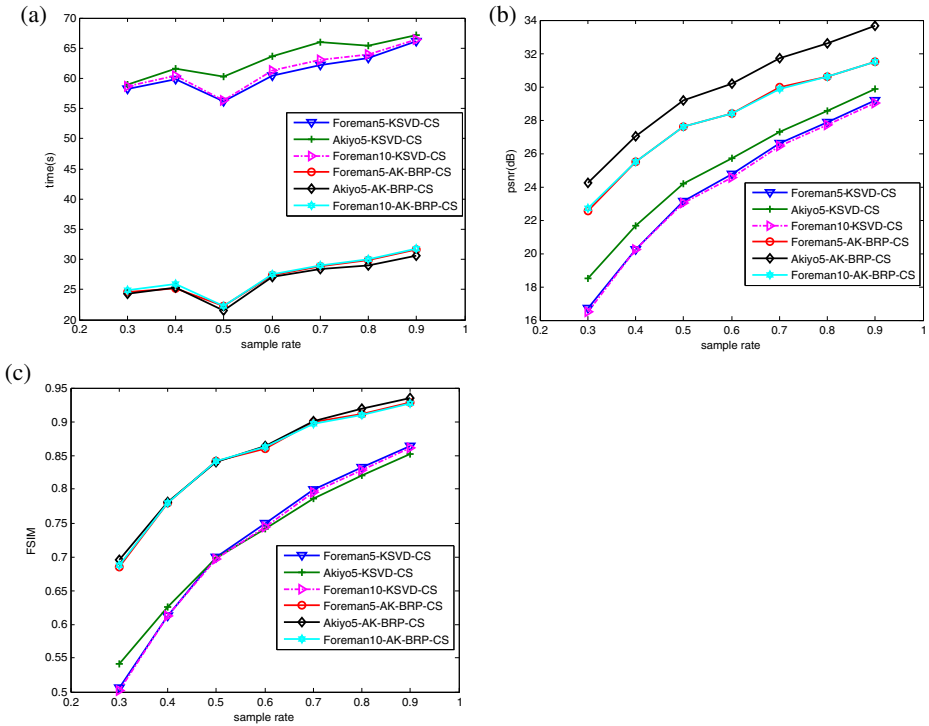
To evaluate the reconstruction quality of each algorithm, PSNR (Peak Signal to Noise Ratio, unit: dB) is used to evaluate the objective image quality. Besides, a recently proposed powerful perceptual quality metric FSIM (Feature SIMilarity) [23] is also calculated to evaluate the visual quality. It has been recognized that the higher FSIM value means the better visual quality. Comparisons of execution time (unit: s), PSNR and FSIM for five gray test frames of Foreman in the case of 0.4 to 0.6 sampling rate are provided in Tables 1, 2 and 3, respectively. Here, in order to eliminate the randomness, all the values of each test frame are averaged over 10 executions.

Tables 1, 2 and 3 show that compared with KSVD-CS algorithm, our proposed algorithm is not only time saving, but also achieves the highest PSNR and FSIM, which can effectively reduce the reconstruction error. In order to compare the merits and demerits of each algorithm more intuitively, we carry out experiments at different sampling rates ranging from 0.3 to 0.9, and the comparison results are presented in Fig. 2. Wherein, the values of running time, PSNR and FSIM are averaged over 5 or 10 random gray test frames. And two types of videos (Foreman and Akiyo) are used.

From Fig. 2a, we can discover that the running time of our AK-BRP-CS algorithm is much lower than that of the KSVD-CS algorithm for both Foreman and Akiyo videos. The reason is that our method can obtain adaptive sparsity in the sparse coding stage, and only calculates the largest singular value and the corresponding singular vector during each dictionary atom update stage, which accelerates the speed of the algorithm. Moreover, Fig. 2b shows that the reconstruction performances of the two algorithms are not very good for low sampling rate. However, with the increase of sampling rate, the two algorithms' reconstruction results both have a significant facelift. It is clear that our method improves the CS reconstruction PSNR remarkably compared to KSVD-CS. Figure 2c once again demonstrates that the proposed algorithm has better visual quality than the KSVD-CS algorithm.

**Table 3** FSIM Comparison with Various CS Methods Based on Different Sparsifying Basis

Ratio	Algorithms	foreman1	foreman6	foreman10	foreman15	foreman23	Average
r = 0.4	<b>KSVD-CS</b>	0.6115	0.62452	0.65486	0.60564	0.56398	0.61210
	<b>Proposed</b>	0.78094	0.78564	0.78872	0.7692	0.77142	<b>0.77918</b>
r = 0.5	<b>KSVD-CS</b>	0.70444	0.71626	0.74728	0.67614	0.65306	0.69944
	<b>Proposed</b>	0.85106	0.83964	0.84246	0.84014	0.833	<b>0.84126</b>
r = 0.6	<b>KSVD-CS</b>	0.7334	0.76426	0.78432	0.73952	0.7227	0.74884
	<b>Proposed</b>	0.87262	0.85916	0.8696	0.85086	0.84936	<b>0.86032</b>



**Fig. 2** Comparison of reconfiguration performance of KSVD-CS and AK-BRP-CS algorithms. **a** Comparison of CS reconstruction running time; **b** comparison of CS reconstruction PSNR; **c** comparison of CS reconstruction FSIM

Figure 2a-c also show that there is negligible difference between performances of 5 and 10 test frames. As a result, it is reasonable to consider 5 random test frames and the performance is acceptable. In addition, since for both Foreman and Akiyo video sequences, AK-BRP-CS algorithm is superior to KSVD-CS, only results on “Foreman” will be shown in the following experiments.

Specifically, some visual results of the recovered 6th frame of “Foreman” by the two algorithms are presented in Fig. 3. Here, the sampling rate is equal to 0.4.

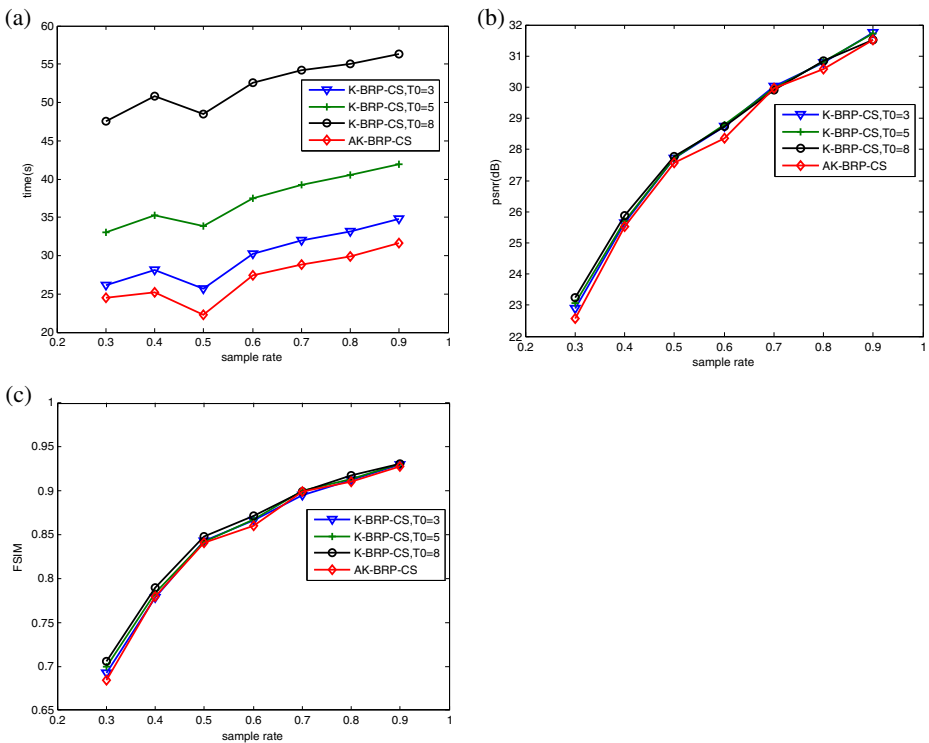


**Fig. 3** Visual quality comparison of CS recovery on the 6th frame with subrate = 0.4. From left to right: the KSVD-CS recovered frame; the AK-BRP-CS recovered frame

Obviously, the proposed algorithm shows much clearer and better visual results than the KSVD-CS algorithm in the same experimental conditions. And the results fully reflect the superiority of our algorithm. However, because of the randomness and adaptivity of our algorithm, there is some poor local performance of AK-BRP-CS, e.g. the top-left corner of the AK-BRP-CS recovered frame.

### 5.3 Effect of sparse coding stage

The superior performance of the proposed algorithm has been adequately demonstrated in the previous section. In this sub-section, we will verify the influences of our adaptive sparsity constraint on the algorithm performance. To make a better comparison, we change the sparse coding stage of the proposed AK-BRP dictionary learning algorithm by using the sparse coding method of the conventional KSVD algorithm, forming the K-BRP dictionary learning algorithm. In our comparison experiments, the K-BRP dictionary and the AK-BRP dictionary are respectively selected as the sparsifying basis of CS, and the OMP reconstruction algorithm is used to restore each test frame. For ease of the elaboration, the CS based on these two dictionaries are credited as K-BRP-CS and AK-BRP-CS, respectively. Figure 4 shows the comparison of reconfiguration performance of K-BRP-CS, whose sparsity is respectively set to 3, 5 and 8, and AK-BRP-CS algorithms.



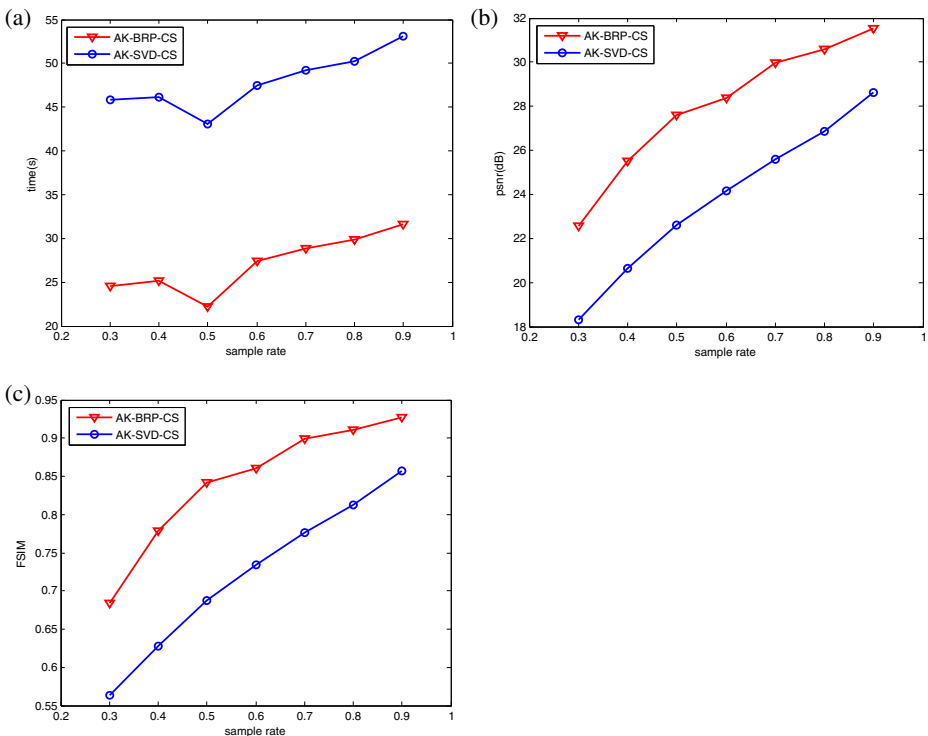
**Fig. 4** Comparison of reconfiguration performance of K-BRP-CS and AK-BRP-CS algorithms in the case of different sparsity. **a** Comparison of CS reconstruction running time; **b** comparison of CS reconstruction PSNR; **c** comparison of CS reconstruction FSIM

Wherein, the values of running time, PSNR and FSIM are averaged over the five gray test frames of “Foreman”, whose values are all averaged over 10 executions.

As can be seen from Fig. 4a, with the increase of the sparsity, the running time of the K-BRP-CS algorithm increases progressively. Especially, when the sparsity is 3, the running time of K-BRP-CS algorithm is close to that of AK-BRP-CS algorithm, which means that the adaptive selection of sparsity in each iteration of our proposed algorithm is less than 3. In addition, Fig. 4b, c both illustrate that the change of sparsity has little effect on the reconstruction performance of K-BRP-CS algorithm, since the CS reconstruction PSNR and FSIM are almost unanimous in the four cases. To sum up, our AK-BRP-CS algorithm simultaneously ensures the reconstruction accuracy and selects the smaller sparsity in every iteration step to decrease the running time, which fully reflects the feasibility of the designed adaptive sparse coding mode.

#### 5.4 Effect of dictionary update stage

In this sub-section, the superiority of the proposed dictionary learning algorithm in the dictionary update stage will be further certificated by the simulation results. In order to make a better comparison, we improve the dictionary update mode of the traditional K-SVD algorithm by using the proposed adaptive sparse coding method, forming an adaptive K-SVD (AK-SVD) dictionary learning algorithm. Similar to Sub-section 5.3, the CS based on AK-SVD dictionary also be credited as AK-SVD-CS for conveniently describing. The comparisons of reconstruction



**Fig. 5** Comparison of reconfiguration performance of AK-SVD-CS and AK-BRP-CS algorithms. **a** Comparison of CS reconstruction running time; **b** comparison of CS reconstruction PSNR; **c** comparison of CS reconstruction FSIM

property of AK-SVD-CS and AK-BRP-CS algorithms are presented in Fig. 5. Wherein, the values of running time, PSNR and FSIM are averaged over the five gray test frames of “Foreman”, whose values are all averaged over 10 executions.

Obviously, compared with the AK-SVD-CS algorithm, our AK-BRP-CS algorithm saves as much as half the running time, and improves the CS reconstruction PSNR and FSIM about 3–5 dB, 0.07–0.15 respectively, when the sampling rate among 0.3 to 0.9. The reason is that the AK-BRP-CS algorithm only calculates the largest singular value and the corresponding singular vector in the dictionary update stage by exploiting the bilateral random projections (BRP) method, and results in a great reduction in the computation cost. In addition, due to the stronger robustness of the BRP method in comparison with SVD decomposition, our algorithm can improve the reconstruction accuracy to some extent. Above all, the way of updating the dictionary in this paper is more practical.

## 6 Conclusion

We present a new method, named adaptive K-BRP (AK-BRP), for learning patch-sparse adaptive redundant dictionaries to improve the accuracy of CS reconstruction. The proposed dictionary learning algorithm associates the sparsity upper-bound with the iterated updated dictionary in the sparse coding stage and adopts bilateral random projections to update each dictionary atom in the following dictionary update stage. A great deal of comparative experiments show that our proposed algorithm achieves significant sparse representation performance improvements over the traditional K-SVD dictionary learning algorithm.

**Acknowledgments** This work was supported in part by National Natural Science Foundation of China (Granted No. 61070234, 61071167, 61373137, 61501251), university graduate student research innovation project of Jiangsu province in 2014 (Granted NO. KYZZ\_0233) and in 2015 (Granted NO. KYZZ15\_0235) and the NUPTSF (Granted No. NY214191).

## References

1. Aharon M, Elad M, Bruckstein A (2006) K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans Signal Process* 54(11):4311–4322
2. Anaraki PF, Hughes SM (2013) Compressive K-SVD. In: *Proceedings of the 2013 I.E. International Conference on Acoustics, Speech, and Signal Processing*. Vancouver, Canada: IEEE, 5469–5473
3. Bahrapour S, Nasrabadi NM, Ray A, and Jenkins WK (2015) “Multimodal task-driven dictionary learning for image classification,” arXiv: 1502.01094
4. Candès E, Tao T (2006) Near optimal signal recovery from random projections: universal encoding strategies [J]. *IEEE Trans Inf Theory* 52(12):5406–5425
5. Candès EJ, Wakin MB (2008) An introduction to compressive sampling. *IEEE Signal Process Mag* 25(2):21–30
6. Tianyi Zhou and Dacheng Tao (2012) Bilateral random projections, *Proceedings of the 2012 I.E. International Symposium on Information Theory*, pp. 1286–1290
7. Tianyi Zhou and Dacheng Zhao (2011) GoDec: Randomized low-rank and sparse matrix decomposition in noisy case, *Proceedings of the 28th International Conference on Machine Learning*, pp. 33–40
8. Delgado KK, Murray JF, Rao BD (2003) Dictionary learning algorithms for sparse representation. *Neural Comput* 15:349–396

9. Donoho DL (2006) Compressed sensing. *IEEE Trans Inf Theory* 52(4):1289–1306
10. Donoho DL, Huo XM (2001) Uncertainty principles and ideal atomic decomposition. *IEEE Trans Inf Theory* 47(7):2845–2862
11. Donoho DL, Tsaig Y (2006) Extensions of compressed sensing [J]. *Signal Process* 86(3):533–548
12. Donoho DL, Tsaig Y (2008) Fast solution of  $l_1$ -norm minimization problems when the solution may be sparse. *IEEE Trans Inf Theory* 54(11):4789–812
13. Elad M, Aharon M (2006) Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans Signal Process* 15(12):3736–3745
14. Liu W, Yu Z, Yang M, Lu L, Zou Y (2015) Joint kernel dictionary and classifier learning for sparse coding via locality preserving K-SVD, in: *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, pp.1-6
15. Liu XM, Zhai DM, Zhao DB, Gao W (2013) Image super-resolution via hierarchical and collaborative sparse representation. In: *Proceedings of the 2013 Data Compression Conference*. Snowbird, USA: IEEE, 93–102
16. Mailhe B, Barchiesi D, Plumbley MD (2012) INK-SVD: learning incoherent dictionaries for sparse representation. In: *Proceedings of the 2012 International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Kyoto, Japan, USA: IEEE, 3573–3576
17. Protter M, Elad M (2009) Image sequence denoising via sparse and redundant representations. *IEEE Trans Image Process* 18(1):27–36
18. Ravishanker S, Bresler Y (2011) MR image reconstruction from highly undersampled K-space data by dictionary learning. *IEEE Trans Med Imaging* 30(5):1028–1041
19. ROWEISS (1998) EM algorithms for PCA and SPCA [C] // *Proceedings of the 1997 Conference on Advances in Neural Information Processing System*. Cambridge: Press, 626-632
20. Rubinstein R, Bruckstein A, Elad M (2010) Dictionaries for sparse representation modeling. *Proc IEEE* 98(6):1045–1057
21. Rubinstein R, Zibulevsky M, Elad M (2010) Double sparsity: learning sparse dictionaries for sparse signal approximation. *IEEE Trans Signal Process* 58(3):1553–1564
22. Zhang Q and Li B (2010) “Discriminative K-SVD for dictionary learning in face recognition,” in *Proc. IEEE Conf. CVPR*, pp. 2691–2698
23. Zhang L, Zhang L, Mou X, Zhang D (2012) FSIM: a feature SIMilarity index for image quality assessment. *IEEE Trans Image Process* 20(8):2378–2386
24. Zhang J, Zhao C, Zhao D et al (2014) Image compressive sensing recovery using adaptively learned sparsifying basis via  $l_0$  minimization [J]. *Signal Process* 103(10):114–126
25. Zheng H, Tao D (2015) Discriminative dictionary learning via fisher discrimination K-SVD algorithm. *Neurocomputing* 162:9–15
26. Zhou FF and Li L (2015) A novel K-RBP dictionary learning algorithm for video image sparse representation, *Journal of Computational Information Systems*, vol. 11 (1), pp.1-11



**Yang Qian** is a postgraduate in School of Science, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu, China. Her research interests include video compressed sensing, machine learning and nonlinear analysis.



**Lei Li** is currently a Professor in Center for Visual Cognitive Computation and Its Application, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu, China. His research interests include data mining, machine learning, compressed sensing, support vector machine, nonlinear analysis, etc.



**Zhenzhen Yang** is currently a Professor in Center for Visual Cognitive Computation and Its Application, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu, China. Her research interests include machine learning, compressed sensing, support vector machine, nonlinear analysis.





**Feifei Zhou** is a postgraduate in School of Science, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu, China. Her research interests include compressed sensing, machine learning and nonlinear analysis.