Single Image Super Resolution Algorithm with a New Dictionary Learning Technique K-Eigen Decomposition

Yingyue Zhou(✉), Hongbin Zang, Su Xu, and Hongying Zhang

Special Environment Robot Technology Key Laboratory of Sichuan Province,
School of Information Engineering,
Southwest University of Science and Technology,
Mianyang 621010, Sichuan, People's Republic of China
zhouyingyue@swust.edu.cn

Abstract. In this paper, we propose an algorithm to improve some important details of sparse representation based image super resolution (SR) framework. Firstly, a new dictionary learning technique K-Eigen decomposition (K-EIG) is proposed. It improves the classical K-SVD algorithm in dictionary atom updating. K-EIG accelerates the learning process and keeps the similar performance of the learned dictionary. Secondly, image patch classification and edge patches extension are integrated into the SR framework. Two over-complete dictionary-pairs are trained based on K-EIG. In reconstruction, the input low resolution (LR) image is split into patches and each one is classified. The patch type decides which dictionary-pair is chosen. Then the sparse representation coefficient of the LR signal is inferred and the corresponding high resolution (HR) patch can be reconstructed. Experimental results prove that our algorithm can obtain competitive SR performance when compared with some classical methods. Besides, the time-consuming of dictionary-pair learning is lower.

Keywords: Image super-resolution · Sparse representation · Dictionary learning

1 Introduction

Super resolution (SR) image reconstruction refers to a signal processing approach to obtain a high resolution (HR) image from observed single or multiple low resolution (LR) image(s) [1]. For the single image SR reconstruction problem, simple interpolation methods like "bilinear" or "bicubic" interpolation cannot do a good job because only the neighbor pixels are used to estimate the missing information. The enlarged image is blurry especially in the regions existing edges and textures. Some researchers [2, 3] merged the natural image priors into the traditional interpolation methods and generated the image with sharper edges. But these methods are unstable and sometimes easy to produce artifacts. Another kind of methods employs the machine learning techniques which are very popular nowadays. Freeman et al. [4, 5] first proposed this idea and named it as "example-based super-resolution". They estimated the missing details with the help of abundant example-pairs extracted from an external training set.

Y.-J. Zhang (Ed.): ICIG 2015, Part III, LNCS 9219, pp. 369–381, 2015.
DOI: 10.1007/978-3-319-21969-1_32

The algorithm attempts to learn the co-occurrence prior between LR and HR image patches. Then the prior is used in SR reconstruction. Example-based SR method breaks through the bottleneck of the conventional SR methods and generates more distinct image. Inspired by the manifold learning, neighbor embedding for SR was proposed [6]. In this method, multiple examples can contribute simultaneously to the generation of each HR image patch. In 2008, Yang et al. [7] employed "sparse coding" to represent an image patch sparsely over an example set. Moreover, the connection between LR and HR example is replaced by the same sparse representation coefficient over the LR or HR example set. Then, Wang [8] and Yang [9] successively used a pair of learned compact over-complete dictionary to replace the two huge example sets. This change speeds up the process of sparse coding greatly. However, it is very time-consuming to learn the over-complete dictionary-pair. So, some researchers try to accelerate the learning process [10–12]. In recent years, Dong et al. [13] proposed a nonlocal centralized sparse representation technique and Peleg et al. [14] presented a statistical prediction model based on sparse representations, which all generates the better SR images.

Here, we propose a fast dictionary learning technique K-Eigen Decomposition (K-EIG) and employ it into the sparse representation based SR framework. In addition, we merge the classification of image patches and the extension of edge patches skillfully into the whole algorithm to promote the quality of the SR images. The outline of this paper is as follows. In Sect. 2, we analyze the essence of the sparse representation based image SR methods. Then, the details of our proposed algorithm are presented in Sect. 3. In Sect. 4, we show the simulations and results. Conclusion is given in Sect. 5.

2 Reviews and Analysis

Let $\boldsymbol{I}_l$ be the given LR image and $\boldsymbol{I}_h$ be the corresponding HR image which needs to be estimated. Usually, the degradation model can be written as:

$$\boldsymbol{I}_l = (\boldsymbol{I}_h * f) \downarrow s, \tag{1}$$

where f is a blurring kernel, $*$ is the convolution operation and $\downarrow s$ means down-sampling with a factor s. In addition to the prior degradation model, an external training set $\boldsymbol{TS}$ containing various HR images $\boldsymbol{T}_h$ can be used. The objective of the example-based image SR methods is to predict the missing details in $\boldsymbol{I}_l$ by "borrowing" information from some similar examples in $\boldsymbol{TS}$ [15] and then reconstruct $\boldsymbol{I}_h$. Next, we make an analysis from the following three aspects for the essence of sparse representation based image SR methods. Meanwhile, some representative prior work is also reviewed.

The first aspect is how to build the link between LR and HR image patches based on $\boldsymbol{TS}$ through learning. The link is a learned prior for SR reconstruction. With the degradation model, the LR version $\boldsymbol{T}_l$ for each $\boldsymbol{T}_h$ can be formed by $\boldsymbol{T}_l = (\boldsymbol{T}_h * f) \downarrow s$. For the sampling convenience, $\boldsymbol{T}_l$ is enlarged to be with the same size of $\boldsymbol{T}_h$ by interpolation. Then, a number of image patch-pairs $(\boldsymbol{Tp}_l, \boldsymbol{Tp}_h)$ are sampled from every image-pair $(\boldsymbol{T}_l,\boldsymbol{T}_h)$, representing the LR and HR version of the same region respectively. In the sparse representation based image SR methods, an effective link is through the common sparse representation coefficient [7–10]. Specifically speaking, a $\boldsymbol{Tp}_l$ can be represented by a sparse vector over a LR over-complete dictionary $\boldsymbol{D}_l$ and at the same time the corresponding $\boldsymbol{Tp}_h$ owns the same sparse vector over a HR dictionary $\boldsymbol{D}_h$. Therefore, a key problem is how to construct the over-complete dictionary-pair $(\boldsymbol{D}_l, \boldsymbol{D}_h)$ to meet the requirement. The second aspect is how to express the LR and HR image patches. In [5], the authors pointed out that the highest spatial-frequency components of the LR image are most important in predicting the extra details. Usually, there are three ways to get them, the high-pass filter [4, 5, 10], the image primitives [15, 16] and the low-order derivatives [6, 9]. For each LR image patch, a signal vector $\boldsymbol{v}_l$ can be formed from its highest spatial-frequency components. For a HR image patch, its mean value or LR version is often subtracted and then $\boldsymbol{v}_h$ is formed. These signal vector-pairs $(\boldsymbol{v}_l, \boldsymbol{v}_h)$ represent the raw image patch-pairs. The third aspect is how to reconstruct $\boldsymbol{I}_h$ from $\boldsymbol{I}_l$. Usually, the input $\boldsymbol{I}_l$ is scaled-up to the size of $\boldsymbol{I}_h$ by interpolation. Since the SR scheme works on the patches, $\boldsymbol{I}_l$ is broken into patches in advance. Then, the signal vector $\boldsymbol{v}_l$ of each patch $\boldsymbol{Ip}_l$ is calculated and its sparse representation vector $\boldsymbol{\alpha}$ is inferred over $\boldsymbol{D}_l$. Depending on the link learned in training, a HR version $\hat{\boldsymbol{I}}\boldsymbol{p}_h$ can be estimated. Through the rational combination such as averaging the multiple estimated values for each pixel, a HR image can be obtained.

3 The Details of Our Proposed SR Method

3.1 A New Dictionary Learning Algorithm K-EIG

Assume that the given set of signal examples is $\boldsymbol{Y} = \{\boldsymbol{y}_i\}_{i=1}^{M}$. $\boldsymbol{y}_i(\boldsymbol{y}_i \in \mathbb{R}^n)$ is a signal example and M is the number of signal examples. According to the theory of signal sparse representation [17], each $\boldsymbol{y}_i$ can be represented approximately by a sparse vector $\boldsymbol{\alpha}_i(\boldsymbol{\alpha}_i \in \mathbb{R}^L)$ over an over-complete dictionary $\boldsymbol{D}$, $\boldsymbol{D} \in \mathbb{R}^{n \times L}(L > n)$. Let $\boldsymbol{A}$ be a sparse vector set $\boldsymbol{A} = \{\boldsymbol{\alpha}_i\}_{i=1}^{M}$. Note that $\boldsymbol{Y}$ and $\boldsymbol{A}$ can be also treated as the matrices. Then, the problem of dictionary learning can be described as:

$$\min_{\boldsymbol{D},\boldsymbol{A}} \{||\boldsymbol{Y} - \boldsymbol{DA}||_F^2\} \; s.t. \forall i, \; ||\boldsymbol{\alpha}_i||_0 \leq T_0, \tag{2}$$

where $\| \boldsymbol{\alpha}_i \|_0$ stands for the count of nonzero entries in $\boldsymbol{\alpha}_i$, T_0 is the maximum allowed number of nonzero entries in $\boldsymbol{\alpha}_i$. By now K-SVD [12] is the most prevalent dictionary learning algorithm [18–20]. It alternates between sparse coding of signal vectors and updating the dictionary atoms to get the solution. The creativity of K-SVD is that it updates one column in $\boldsymbol{D}$ at a time. All columns in $\boldsymbol{D}$ except one atom d_k is fixed. Besides, the kth row in $\boldsymbol{A}$, denoted as α_T^k, is also put in question. So, the penalty term can be rewritten as:

$$||\boldsymbol{Y} - \boldsymbol{DA}||_F^2 = ||(\boldsymbol{Y} - \sum_{j \neq k} \boldsymbol{d}_j \boldsymbol{\alpha}_T^j) - \boldsymbol{d}_k \boldsymbol{\alpha}_T^k||_F^2 = ||\boldsymbol{E}_k - \boldsymbol{d}_k \boldsymbol{\alpha}_T^k||_F^2. \tag{3}$$

$\boldsymbol{E}_k$ stands for the representation error matrix based on the current $\boldsymbol{D}$ and $\boldsymbol{A}$ when $\boldsymbol{d}_k$ is removed. Note that some signal examples do not use $\boldsymbol{d}_k$, so $\boldsymbol{E}_k$ should be restricted as $\boldsymbol{E}_k^R$ by choosing only the related columns. In order to use $\boldsymbol{E}_k^R$ to find the unknown $\tilde{\boldsymbol{d}}_k$ and $\tilde{\boldsymbol{\alpha}}_T^k$ for minimizing $||\boldsymbol{E}_k^R - \boldsymbol{d}_k\boldsymbol{\alpha}_T^k||_F^2$, the singular value decomposition (SVD) of $\boldsymbol{E}_k^R$ is employed:

$$\boldsymbol{E}_k^R = \mathbf{USV}^T,\ \tilde{\boldsymbol{d}}_k = \mathbf{U}(:,1),\ \tilde{\boldsymbol{\alpha}}_T^k = \mathbf{V}^T(:,1) \times \mathbf{S}(1,1). \tag{4}$$

After updating dictionary atoms one by one, a new $\boldsymbol{D}$ can be obtained. Then, go back to the stage of sparse coding. A satisfactory dictionary fitting $\boldsymbol{Y}$ can be generated after several alternations. Unfortunately, the operation of $\boldsymbol{E}_k^R$'s SVD is time-consuming since the column number of $\boldsymbol{E}_k^R$ is usually large. It causes the high computational cost. Rubinstein et al. [11] offered an approximate K-SVD (AK-SVD) algorithm to accelerate K-SVD. However, it sacrifices the performance of the learned dictionary to promote the training speed. Next, we propose a new algorithm K-EIG to accelerate the learning process and try to keep the similar performance of the learned dictionary. The details of derivation are as follows.

For the convenience of derivation, the target function $||\boldsymbol{E}_k^R - \boldsymbol{d}_k\boldsymbol{\alpha}_T^k||_F^2$ is denoted as $||\boldsymbol{E} - \boldsymbol{d\alpha}||_F^2$. Then, we expand $||\cdot||_F^2$:

$$\begin{aligned} fu = \ & ||\boldsymbol{E} - \boldsymbol{d\alpha}||_F^2 = \sum_j \sum_i (d(i)\alpha(j) - E(i,j))^2 \\ & = \sum_j \sum_i (d(i)^2\alpha(j)^2 - 2d(i)\alpha(j)E(i,j) + E(i,j)^2). \\ & = \sum_j [(\sum_i d(i)^2)\alpha(j)^2 - 2(\sum_i d(i)E(i,j))\alpha(j) + \sum_i E(i,j)^2]. \end{aligned} \tag{5}$$

Note that $\sum_i d(i)^2 = 1$. Besides, we can ignore $\sum_i E(i,j)^2$ because it can be treated as a constant. Therefore, fu turns to:

$$fu = \sum_j [\alpha(j)^2 - 2(\sum_i d(i)E(i,j))\alpha(j)]. \tag{6}$$

The formula in $[\cdot]$ can be considered as a form of quadratic equation about $\alpha(j)$. So, the minimum value of fu is reached at the bottom of the quadratic curve when $\alpha(j) = \sum_i d(i)E(i,j)$. And now:

$$fu = -\sum_j (\sum_i d(i)E(i,j))^2. \tag{7}$$

Thus $\alpha(j)$ is eliminated from fu and we can solve $\boldsymbol{d}$ directly under the normalization constraint with the help of Lagrange multiplier method. The Lagrange function is:

$$Lf = -\sum_j \Big(\sum_i d(i)E(i,j)\Big)^2 + \lambda\Big(\sum_i d(i)^2 - 1\Big). \tag{8}$$

Let $\partial Lf/\partial d(m) = 0$ where $d(m)$ is the mth element in $\boldsymbol{d}$. Then we can obtain:

$$\begin{aligned} \partial Lf/\partial d(m) &= -\sum_j \Big(\sum_i d(i)E(i,j)\Big)E(m,j) + \lambda d(m) = 0 \\ &\Rightarrow \sum_i d(i)\sum_j E(i,j)E(m,j) - \lambda d(m) = 0, \quad m = 1,2,\ldots,n. \end{aligned} \tag{9}$$

Let $\boldsymbol{R}$ be the autocorrelation matrix of $\boldsymbol{E}$, i.e. $\boldsymbol{R} = \boldsymbol{E}\cdot\boldsymbol{E}^T$. So, $\sum_j E(i,j)E(m,j) = R(i,m)$. Then, Eq. (9) turns to:

$$\begin{aligned} &\sum_i R(i,m)d(i) - \lambda d(m) = 0,\ m = 1,2,\ldots,n \\ &\Rightarrow R(:,m)^T\boldsymbol{d} = \lambda d(m),\ m = 1,2,\ldots,n \Rightarrow \boldsymbol{R}^T\boldsymbol{d} = \lambda\boldsymbol{d}. \end{aligned} \tag{10}$$

Obviously, the solution of the above problem can be obtained by calculating the eigenvector of $\boldsymbol{R}^T$. We choose the eigenvector corresponding to the largest eigenvalue as the optimal solution of the problem. Then, use $\boldsymbol{\alpha} = \boldsymbol{d}^T\boldsymbol{E}$ to get $\boldsymbol{\alpha}$. Thus, we replace the implementation of $\boldsymbol{E}$'s SVD with $\boldsymbol{R}^T$'s eigen-decomposition and name the whole algorithm as K-Eigen decomposition. Compared with K-SVD, the advantage of K-EIG is computational simplicity. The reason is as follows. In K-SVD, the object of SVD is the matrix $\boldsymbol{E}_k^R$ with the size $n \times M_k$, where M_k represents the number of signals using $\boldsymbol{d}_k$. Usually, M_k is related to the total number of input signals M. With the increasing of M, the column number of $\boldsymbol{E}_k^R$ is larger. It results in poor computational efficiency due to the slow implementation of $\boldsymbol{E}_k^R$'s SVD. Whereas, K-EIG implements eigen-decomposition of $\boldsymbol{R}^T$. $\boldsymbol{R}^T$ is a square matrix of size $n \times n$. It is much smaller than $\boldsymbol{E}_k^R$ since $n \ll M_k$. So, the implementation of $\boldsymbol{R}^T$'s eigen-decomposition is much faster. It makes K-EIG more time-saving.

As described in Sect. 2, based on the signal vector-pairs $(\boldsymbol{v}_l, \boldsymbol{v}_h)$ sampled from $(\boldsymbol{T}_l, \boldsymbol{T}_h)$, we train the dictionary-pair $(\boldsymbol{D}_l, \boldsymbol{D}_h)$ to satisfy the following formula:

$$\begin{cases} \min\limits_{\boldsymbol{D}_l,\boldsymbol{\alpha}} \| \boldsymbol{v}_l^{(i)} - \boldsymbol{D}_l\cdot\boldsymbol{\alpha}^{(i)} \|_2^2 \ s.t.\ \| \boldsymbol{\alpha}^{(i)} \|_0 \le T_0 \\ \min\limits_{\boldsymbol{D}_h,\boldsymbol{\alpha}} \| \boldsymbol{v}_h^{(i)} - \boldsymbol{D}_h\cdot\boldsymbol{\alpha}^{(i)} \|_2^2 \ s.t.\ \| \boldsymbol{\alpha}^{(i)} \|_0 \le T_0 \end{cases}, \quad \forall i \in \{1,2,\ldots,M\}. \tag{11}$$

In order to get $(\boldsymbol{D}_l, \boldsymbol{D}_h)$, we first generate $\boldsymbol{D}_l$ based on $\{\boldsymbol{v}_l^{(1)}, \boldsymbol{v}_l^{(2)}, \ldots, \boldsymbol{v}_l^{(M)}\}$ by solving:

$$\boldsymbol{D}_l, \{\boldsymbol{\alpha}^{(i)}\} = \underset{\boldsymbol{D}_l, \{\boldsymbol{\alpha}^{(i)}\}}{\arg\min} \sum_{i=1}^{M} \| \boldsymbol{v}_l^{(i)} - \boldsymbol{D}_l \cdot \boldsymbol{\alpha}^{(i)} \|_2^2, \ s.t. \ \| \boldsymbol{\alpha}^{(i)} \|_0 \le T_0, \quad \forall i \in \{1, 2, \ldots, M\}. \tag{12}$$

K-EIG can serve the above problem and we get $\boldsymbol{D}_l$ and $\{\boldsymbol{\alpha}^{(i)}\}$. Then, $\{\boldsymbol{\alpha}^{(i)}\}$ and $\{\boldsymbol{v}_h^{(1)}, \boldsymbol{v}_h^{(2)}, \ldots, \boldsymbol{v}_h^{(M)}\}$ both guide the computation of $\boldsymbol{D}_h$ directly as the way in [10]. Let $\boldsymbol{V}_h = [\boldsymbol{v}_h^{(1)}, \boldsymbol{v}_h^{(2)}, \ldots, \boldsymbol{v}_h^{(M)}]$ and $\boldsymbol{A} = [\boldsymbol{\alpha}^{(1)}, \boldsymbol{\alpha}^{(2)}, \ldots, \boldsymbol{\alpha}^{(M)}]$, then $\boldsymbol{D}_h$ can be obtained by $\boldsymbol{D}_h = \boldsymbol{V}_h \cdot \boldsymbol{A}^T \cdot (\boldsymbol{A}\boldsymbol{A}^T)^{-1}$. Now, $(\boldsymbol{D}_l, \boldsymbol{D}_h)$ can meet the requirement of common sparse representation coefficient. Compared with the joint dictionary training method used in [9], the direct dictionary-pair construction method employing K-EIG is fast and effective.

3.2 Sparse Representation Based SR Framework with Example Classification and Extension

Considering the types of image patches, we do an important work before training. That is classification for the sampled image patches. Each patch-pair $(\boldsymbol{Tp}_l, \boldsymbol{Tp}_h)$ will be put into a classifier and then marked by a label "smooth" or "edge". We employ the sorted quadrant median vector (SQMV) scheme proposed in [21] to classify every $\boldsymbol{Tp}_h$. SQMV is a simple but efficient tool to recognize an image patch belonging to uniform region or edge region. Then, the set of example-pairs is separated into two parts, "smooth" set and "edge" set. In order to increase the number of "edge" patch-pairs before learning, we rotate each HR image $\boldsymbol{T}_h$ by three different angles (90°, 180° and 270°) and three different images are formed. Then, we can generate more "edge" patch-pairs. For "smooth" class, there is no need to increase the patch number. Based on the two sets of signal vector-pairs, two different dictionary-pairs $(\boldsymbol{D}_l^{smooth}, \boldsymbol{D}_h^{smooth})$ and $(\boldsymbol{D}_l^{edge}, \boldsymbol{D}_h^{edge})$ are constructed respectively.

In reconstruction, the input LR image $\boldsymbol{I}_l$ is first interpolated to the objective size and then broken into lots of overlapped patches $\boldsymbol{Ip}_l$ with the size of $w \times w$. For each $\boldsymbol{Ip}_l$, we use the first-order combining second-order derivatives to represent it and form a signal vector $\boldsymbol{v}_l$. Through labeling $\boldsymbol{Ip}_l$ by SQMV scheme, we choose the appropriate dictionary-pair $(\boldsymbol{D}_l^{smooth}, \boldsymbol{D}_h^{smooth})$ or $(\boldsymbol{D}_l^{edge}, \boldsymbol{D}_h^{edge})$ for SR. A key problem is to infer the sparse representation vector of $\boldsymbol{v}_l$ over $\boldsymbol{D}_l$ by $\hat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}}{\arg\min} \| \boldsymbol{v}_l - \boldsymbol{D}_l \cdot \boldsymbol{\alpha} \|_2^2 \ s.t. \ \| \boldsymbol{\alpha} \|_0 \le T_0$. Orthogonal matching pursuit (OMP) [22] or convex relaxation techniques [17] can be used to solve the problem. After getting $\hat{\boldsymbol{\alpha}}$ for $\boldsymbol{Ip}_l$, we can reconstruct its HR version by $\hat{\boldsymbol{Ip}}_h = \boldsymbol{Ip}_l + reshape(\boldsymbol{D}_h \cdot \hat{\boldsymbol{\alpha}})$ where $reshape(\cdot)$ means rearranging a vector to a patch. By the simple "averaging" of these overlapped $\hat{\boldsymbol{Ip}}_h$, we get the estimated HR image $\hat{\boldsymbol{I}}_h$. Besides, to satisfy the degradation model, we add an operation of "backprojection" [9] to modify $\hat{\boldsymbol{I}}_h$ and generate the last $\hat{\boldsymbol{I}}_h$.

4 Simulations and Results

4.1 Testing the Performance of K-EIG

We test the performance of K-EIG by a synthetic experiment and compare it with other typical dictionary learning algorithms. A random matrix $\boldsymbol{D}_g$ of size 20 × 50 is generated with i.i.d uniformly distributed entries. Each column is normalized to a unit ℓ_2-norm. Then a number of data signals $\{\boldsymbol{y}_i\}_{i=1}^{M}$ of dimension 20 are produced. Each signal is created by a linear combination of three different atoms in $\boldsymbol{D}_g$, with uniformly distributed i.i.d coefficients in random and independent locations. Then, K-SVD [12], AK-SVD [11] and K-EIG are implemented respectively to obtain three recovered dictionaries $\boldsymbol{D}_r$ based on $\{\boldsymbol{y}_i\}_{i=1}^{M}$ and then we observe which $\boldsymbol{D}_r$ is closest to $\boldsymbol{D}_g$. Authors can see [12] for more details about success ratio computation. In order to test the robustness of algorithm, white Gaussian noise with varying signal-to-noise ratio (SNR) is added to $\{\boldsymbol{y}_i\}_{i=1}^{M}$. Let $M = 1500$ and $T_0 = 3$. Set the number of alternate iteration by 30. The results of atom recovery performance are shown in Table 1. We see that K-SVD obtains the best performance under any SNR situation. The average success ratio of K-EIG is very close to K-SVD. AK-SVD is inferior to the other two algorithms. Besides, we compare the running speed of the three algorithms. We change the input signal number from $M =$ 1000 to $M = 96000$ with interval of 5000 and keep the other parameters unchanged. The running platform is MATLAB 2009 and the computer is equipped with 2.60 GHZ Inter (R) Core(TM) i5 CPU & 4.0 GB memory. Figure 1 shows the time consuming curve. We see that the computation time of K-SVD rises rapidly with the increasing of input signals number. On the contrary, the time consuming of K-EIG and AK-SVD always keeps a low level even in the situation of large M.

Table 1. Comparison of several dictionary learning algorithms in average success ratio of atom recovery

SNR (dB)	K-SVD	AK-SVD	K-EIG
20	92.68 %	91.48 %	92.64 %
15	89.24 %	88.36 %	88.72 %
10	86.44 %	83.28 %	85.92 %
8	70.08 %	66.60 %	69.32 %

4.2 Testing Image SR Performance

In our SR experiments, we use a training set $\boldsymbol{TS}$ containing some HR natural images such as flowers, architectures, animals and human faces. In simulation, we assume that the degradation model $\boldsymbol{I}_l = (\boldsymbol{I}_h * f) \downarrow s$ is operated by the "bicubic filter" with a zoom factor $1/s$. During training, we extract about 8×10^4 image patch-pairs to be the training examples for each class. The patch size is 5 × 5. The column number of $\boldsymbol{D}_l$ or $\boldsymbol{D}_h$ is set by 1000. The number of alternate iteration of K-EIG is set by 40. Five color images are chosen as the test images. They are "lena", "child", "flower", "building", and "zebra". Then, five corresponding LR images are generated by the known

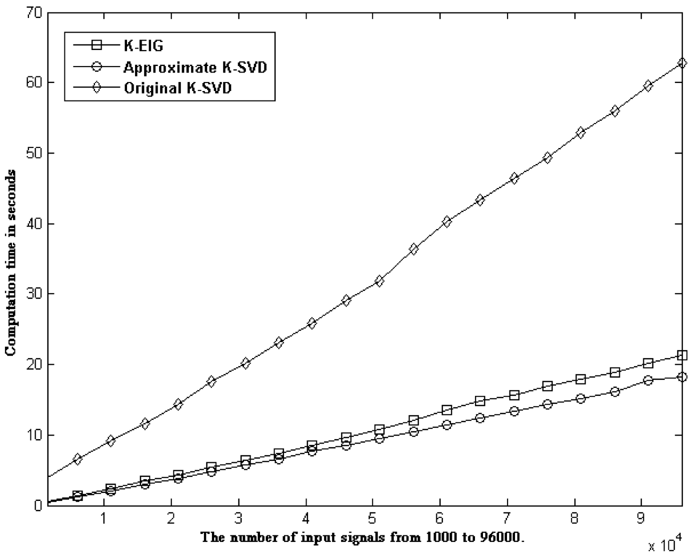

Fig. 1. Time consuming curve of three dictionary learning algorithms

degradation model. Our task is to reconstruct the HR versions of these LR images. As the way in other literatures, we just apply our SR algorithm to the illuminance channel (Y). The other channels (Cb, Cr) are processed by bicubic interpolation simply. Several representative and related image SR methods are also implemented for comparison purpose. They are “bicubic interpolation”, “neighbor embedding [6]”, “Yang’s method [9]” and “Zeyde’s method [10]”. We choose the peak signal-to-noise ratio (PSNR) and the mean structure similarity index measure (MSSIM) [23] as the performance indexes to measure the quality of the reconstructed image.

We show the quantitative SR results in Tables 2 and 3 in case of $s = 2$ and $s = 3$. Clearly, our proposed method achieves the highest PSNRs and MSSIMs for all the test

Table 2. PSNRs (MSSIMs) results of reconstructed images with $s = 2$

Image	s = 2				
	Bicubic interpolation	Neigbor embedding	Yang’s	Zeyde’s	Our algorithm
Lena	35.41 (0.9297)	34.78 (0.9169)	37.24 (0.9436)	36.86 (0.9411)	37.56 (0.9457)
Child	37.06 (0.9513)	35.73 (0.9345)	38.37 (0.9629)	38.07 (0.9606)	38.64 (0.9659)
Flower	30.42 (0.8985)	29.81 (0.8764)	32.55 (0.9301)	32.11 (0.9249)	32.92 (0.9339)
Building	28.81 (0.8525)	27.92 (0.8233)	30.22 (0.8886)	29.91 (0.8817)	30.57 (0.8918)
Zebra	30.68 (0.9084)	29.80 (0.8774)	33.33 (0.9410)	32.86 (0.9352)	33.66 (0.9440)

Table 3. PSNRs (MSSIMs) results of reconstructed images with $s = 3$

Image	s = 3				
	Bicubic interpolation	Neigbor embedding	Yang's	Zeyde's	Our algorithm
Lena	32.11 (0.8798)	31.52 (0.8592)	33.57 (0.8997)	33.42 (0.8973)	33.63 (0.9009)
Child	33.94 (0.9042)	32.73 (0.8760)	34.97 (0.9206)	34.98 (0.9195)	35.18 (0.9225)
Flower	27.32 (0.8027)	26.71 (0.7644)	28.62 (0.8427)	28.50 (0.8388)	28.66 (0.8440)
Building	26.14 (0.7518)	25.54 (0.7208)	26.77 (0.7839)	26.73 (0.7788)	26.85 (0.7861)
Zebra	26.69 (0.7946)	25.57 (0.7383)	28.49 (0.8445)	28.50 (0.8390)	28.78 (0.8484)

images. In order to compare the reconstruction results visually, we present the estimated HR images by nearest interpolation, bicubic interpolation, neighbor embedding, Yang's, Zeyde's and our method in case of $s = 3$ in Figs. 2 and 3. Meanwhile, some details are magnified to clearly display the subtle difference. From them, we see that bicubic interpolation eliminates the sawtooth appearance in the images obtained by

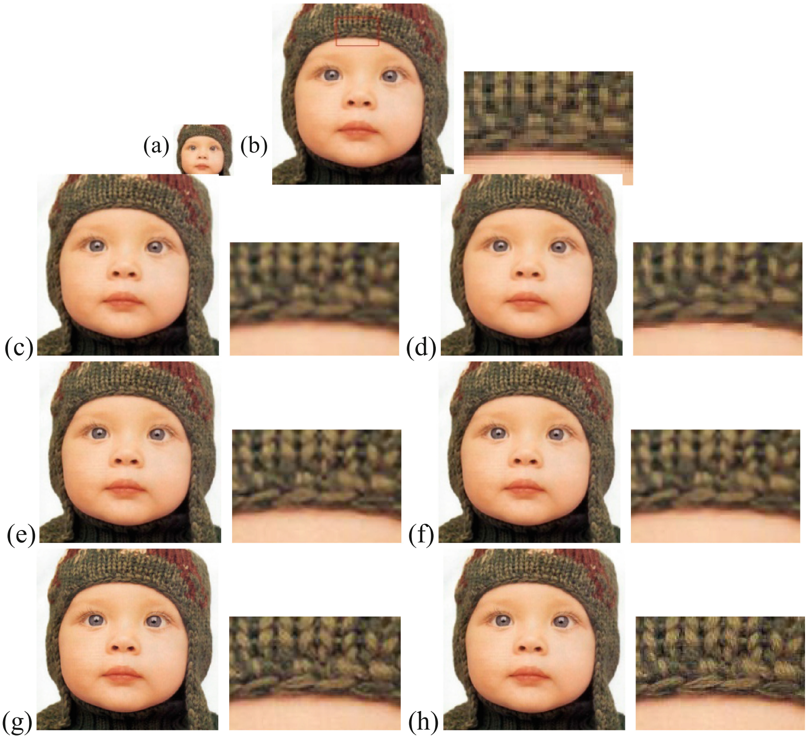

Fig. 2. (a) Input LR 'child' image, (b) estimated HR 'child' by nearest interpolation, (c) estimated HR 'child' by bicubic interpolation, (d) estimated HR 'child' by neigbor embedding, (e) estimated HR 'child' by Yang's method, (f) estimated HR 'child' by Zeyde's, (g) estimated HR 'child' by our method, (h) original 'child' image.

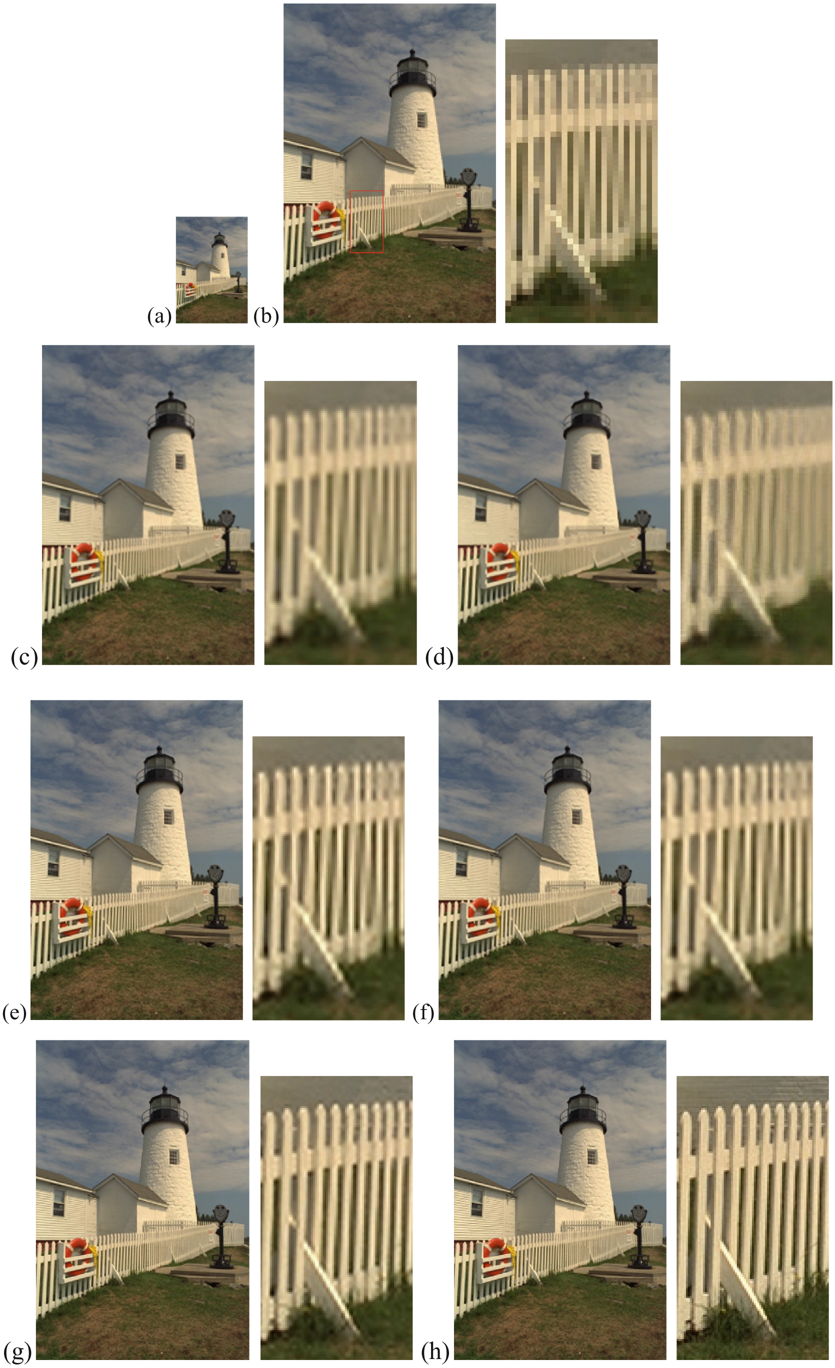

Fig. 3. (a) Input LR 'building' image, (b) estimated HR 'building' by nearest interpolation, (c) estimated HR 'building' by bicubic interpolation, (d) estimated HR 'building' by neigbor embedding, (e) estimated HR 'building' by Yang's method, (f) estimated HR 'building' by Zeyde's, (g) estimated HR 'building' by our method, (h) original 'building' image.

nearest interpolation, but it generates the blurred edges. Neighbor embedding is easy to produce some artifacts. For example, in the part between hat and face in Fig. 2(d), we can observe the discontinuous. Sparse representation based methods generate clearer HR images. Among them, our algorithm obtains the best results which are closest to the original HR images, thanks to the integration of patch classification and edge patches extension into the SR framework. In Fig. 3, we can see obviously that the lattice in Fig. 3(g) is reconstructed better than Fig. 3(b) and (f).

4.3 Testing the Time Performance of Dictionary-Pair Construction

Here, we compare the time performance of the three dictionary-pair construction methods (Yang's, Zeyde's and ours). The platform and computer configuration are the same as description in Sect. 4.1. The CPU running time is presented in Table 4. Obviously, our and Zeyde's method are much faster than Yang's method. Our method is slightly slower than Zeyde's one because our training process includes patch-classification, edge-patches extension and learning of two dictionary-pairs. It is worthy of spending more several minutes to obtain better dictionary-pair.

Table 4. Time comparison of three dictionary-pair constructing methods

Method	Time consuming (in s)
Yang's	10007
Zeyde's	238
Ours	411

5 Conclusion

In this paper, we propose a sparse representation based single image super resolution algorithm with a new dictionary learning technique K-EIG. Our proposed K-EIG is an improved algorithm based on the classical K-SVD algorithm. It replaces the implementation of SVD about the representation error matrix with eigen-decomposition about its autocorrelation matrix. This change overcomes the drawback that the operation of SVD is time-consuming in case of the large number of training examples. We employ a direct dictionary-pair construction method based on K-EIG to accelerate the whole training process. Besides, considering the types of image patches and the number of edge examples, patch classification and edge patches extension are integrated rationally into the SR framework. Extensive experimental results show that our SR algorithm generates better SR images compared with some classical or related methods.

Acknowledgment. This research is supported by the National Natural Science Foundation of China (Grant No.61401379), the General Project of Educational Commission of Sichuan Province in China (Grant No.14ZB0107), the Doctoral Research Fund of Southwest University of

Science and Technology in China (Grant No.13zx7148) and the Key Project of Educational Commission of Sichuan Province in China (Grant No.11ZA130).

References

1. Park, S.C., Park, M.K., Kang, M.G.: Super-resolution image reconstruction: a technical overview. IEEE Signal Process. Mag. **20**(3), 21–36 (2003)
2. Sun, J., Xu, Z., Shum, H.: Image super-resolution using gradient profile prior. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2008)
3. Dai, S., Han, M., Xu, W.: SoftCuts: a soft edge smoothness prior for color image super-resolution. IEEE Trans. Image Process. **18**(5), 969–981 (2009)
4. Freeman, W.T., Pasztor, E.C., Carmichael, O.T.: Learning low-level vision. Int. J. Comput. Vis. **40**(1), 25–47 (2000)
5. Freeman, W.T., Jones, T.R., Pasztor, E.C.: Example-Based super-resolution. IEEE Comput. Graphics Appl. **22**(2), 56–65 (2002)
6. Chang, H., Yeung, D.Y., Xiong, Y.: Super-resolution through neighbor embedding. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 275–282 (2004)
7. Yang, J., Wright, J., Huang, T., et al.: Image super-resolution as sparse representation of raw image patches. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2008)
8. Wang, J., Zhu, S., Gong, Y.: Resolution enhancement based on learning the sparse association of image patches. Pattern Recogn. Lett. **31**(1), 1–10 (2010)
9. Yang, J., Wright, J., Huang, T., et al.: Image super-resolution via sparse representation. IEEE Trans. Image Process. **19**(11), 1–8 (2010)
10. Zeyde, R., Elad, M., Protter, M.: On single image scale-up using sparse-representations. In: Boissonnat, J.-D., Chenin, P., Cohen, A., Gout, C., Lyche, T., Mazure, M.-L., Schumaker, L. (eds.) Curves and Surfaces. Lecture Notes in Computer Science, vol. 6920, pp. 711–730. Springer, Heidelberg (2010)
11. Rubinstein, R., Zibulevsky, M., Elad, M.: Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit. Technical report, Technion-Israel Institute of Technology (2008)
12. Aharon, M., Elad, M., Bruckstein, A.: K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans. Signal Process. **54**(11), 4311–4322 (2006)
13. Dong, W., Zhang, L., Lukac, R., Shi, G.: Nonlocal centralized sparse representation for image restoration. IEEE Trans. Image Process. **22**(4), 1382–1394 (2013)
14. Peleg, T., Elad, M.: A statistical prediction model based on sparse representations for single image super-resolution. IEEE Trans. Image Process. **23**(7), 3085–3098 (2014)
15. Fan, W., Yeung, D.Y.: Image hallucination using neighbor embedding over visual primitive manifolds. In: Proceedings of IEEE Conference on Computer Vision Pattern Recognition (2007)
16. Li, H., Xiong, H., Qian L.: Image super-resolution with sparse representation prior on primitives. In: Proceedings of Visual Communications and Image Processing (2010)
17. Elad, M.: Sparse and Redundant Representations From Theory to Applications in Signal and Image Processing. Springer, New York (2010)

18. Elad, M., Aharon, M.: Image denoising via sparse and redundant representations over learned dictionaries. IEEE Trans. Image Process. **15**(12), 3736–3745 (2006)
19. Zhou, Y., Ye, Z., Xiao, Y.: A restoration algorithm for images contaminated by mixed Gaussian plus random-valued impulse noise. J. Vis. Commun. Image Represent. **24**(3), 283–294 (2013)
20. Bao, G., Ye, Z., Xu, X., et al.: A compressed sensing approach to blind separation of speech mixture based on a two-layer sparsity mode. IEEE Trans. Audio Speech Lang. Process. **21** (5), 899–906 (2013)
21. Lin, C.H., Tsai, J.S., Chiu, C.T.: Switching bilateral filter with a texture/noise detector for universal noise removal. IEEE Trans. Image Process. **19**(9), 2307–2320 (2010)
22. Tropp, J.A.: Greed is good: algorithmic results for sparse approximation. IEEE Trans. Inf. Theory **50**, 2231–2242 (2004)
23. Wang, Z., Bovik, A.C., Sheikh, H.R., et al.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)