

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/267099304>

Fast Single Image Super-Resolution via Self-Example Learning and Sparse Representation

Article in IEEE Transactions on Multimedia · October 2014

DOI: 10.1109/TMM.2014.2364976

CITATIONS

60

READS

240

4 authors, including:



Fangda Guo

Northeastern University (Shenyang, China)

6 PUBLICATIONS 90 CITATIONS

[SEE PROFILE](#)



Hai Yu

Northeastern university, China, Shenyang

83 PUBLICATIONS 1,342 CITATIONS

[SEE PROFILE](#)



Chen Chen

University of North Carolina at Charlotte

146 PUBLICATIONS 3,585 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Deep learning for image classification, action recognition, automatic target detection, remote sensing analysis [View project](#)



PhD openings (with financial support) in Computer Vision and Deep Learning @ University of North Carolina at Charlotte [View project](#)

Fast Single Image Super-Resolution via Self-Example Learning and Sparse Representation

Zhiliang Zhu, *Member, IEEE*, Fangda Guo*, Hai Yu and Chen Chen, *Student Member, IEEE*

Abstract—In this paper, we propose a novel algorithm for fast single image super-resolution based on self-example learning and sparse representation. We propose an efficient implementation based on the K-singular value decomposition (SVD) algorithm, where we replace the exact SVD computation with a much faster approximation, and we employ the straightforward orthogonal matching pursuit algorithm, which is more suitable for our proposed self-example-learning-based sparse reconstruction with far fewer signals. The patches used for dictionary learning are efficiently sampled from the low-resolution input image itself using our proposed sample mean square error strategy, without an external training set containing a large collection of high-resolution images. Moreover, the ℓ^0 -optimization-based criterion, which is much faster than ℓ^1 -optimization-based relaxation, is applied to both the dictionary learning and reconstruction phases. Compared with other super-resolution reconstruction methods, our low dimensional dictionary is a more compact representation of patch pairs and it is capable of learning global and local information jointly, thereby reducing the computational cost substantially. Our algorithm can generate high-resolution images that have similar quality to other methods but with a greater than hundredfold increase in the computational efficiency.

Index Terms—Approximate K-singular value decomposition, sample mean square error, self-example, single image super-resolution, sparse representation.

I. INTRODUCTION

SUPER-resolution (SR) [1] image reconstruction aims to restore original high-resolution (HR) information beyond the Nyquist frequency from single or multiple low-resolution (LR) images based on reasonable assumptions or prior knowledge of the generation model that maps the LR images to the HR image. SR plays a very active and important role in image processing research because it provides solutions that overcome the resolution limitations [2], [3] attributable to low cost digital imaging sensors (such as monitors or mobile phones) and imperfect imaging environments (such as satellites). The conventional approach for synthesizing a new HR image uses one or more LR images. However, if the number of LR images is inadequate, this causes uncertain blurring operations and ill-conditioned registration; thus, SR reconstruction is usually

This research was supported by the National Natural Science Foundation of China (Grant Nos. 61374178 and 61202085), the Liaoning Provincial Natural Science Foundation of China (Grant No. 201202076), the Specialized Research Fund for the Doctoral Program of Higher Education (Grant No. 20120042120010), and the Ph.D. Start-up Foundation of Liaoning Province, China (Grant Nos. 20111001, 20121001, and 20121002).

Zhiliang Zhu, Fangda Guo (corresponding author) and Hai Yu are with the College of Software, Northeastern University, Shenyang, Liaoning, China (e-mail: zzl@mail.neu.edu.cn; gfdyes@gmail.com; yuhai@mail.neu.edu.cn).

Chen Chen is with the Department of Electrical Engineering, The University of Texas at Dallas, Richardson, TX 75080 USA (e-mail: chenchen870713@gmail.com).

considered to be an ill-posed problem [4]-[6]. Current SR reconstruction algorithms can be divided into interpolation [7]-[9], multiple-image SR [10]-[12], and example-based SR [13]-[15]. Some algorithms deliver good performance but they are computationally inefficient, and vice versa. Therefore, a fast SR reconstruction algorithm with reasonably high performance is needed to achieve an acceptable trade off of performance and computational cost.

Interpolation approaches are used to generate HR images from a single LR image with smoothing methods such as bilinear or bicubic interpolation. These approaches are computationally efficient but they often produce overly smooth HR images with ringing and jagged artifacts. Interpolation by exploiting the natural image priors can produce relatively more favorable results. Dai *et al.* [8] obtained the local image patches using foreground and background descriptors before reconstructing the sharp discontinuity between them. Sun *et al.* [9] explored the gradient profile prior of local image structures and applied it to SR. In traditional multiple-image SR reconstruction, an HR image can be obtained from a set of LR images of the same scene with sub-pixel misalignments. However, this method is unsatisfactory because it only obtains a small improvement in resolution, while requiring much more computation time than the simple interpolation method.

In example-based SR reconstruction, the corresponding relationships between LR and HR image patches are learned from a large number of known LR and HR image patch pairs in the training set, before the learned corresponding relationships are used to reconstruct a new HR image. Thus, it is possible that the missing HR image details can be obtained from HR images training set. Yang *et al.* [16] presented a sparse coding model for learning two dictionaries from an external training set for LR and HR images, which allows the LR and HR image patch pairs to share the same sparse representation. Zeyde *et al.* [17] modified Yang *et al.*'s algorithm [16] by including a major simplification of the overall process in terms of both the computational complexity and the algorithm architecture. Zhang *et al.* [18] proposed a sparse representation based SR method by learning a dual dictionary and replacing the sparse recovery step by simple matrix multiplication to reduce the computational complexity. At present, the most common implementation of example-based SR reconstruction [16] generates two dictionaries simply by randomly sampling raw patches from images with similar statistical characteristics in the training set. The performance of this method depends greatly on the number of atoms in the two dictionaries, i.e., the size of a dictionary pair. If the number of atoms in the dictionary pair is reduced, the SR reconstruction result will

degrade remarkably. However, learning a large number of atoms in a dictionary pair requires a very long computation time for the algorithm. In addition, the example-based SR reconstruction method that uses a training set depends greatly on the HR images in the database. Another essential question is whether a global dictionary or dictionary pair exists that can accurately represent all image patches. Learning a training set may allow more image samples to be employed, but the accuracy of the information provided by the database cannot be guaranteed for any given LR image. The reconstructed HR image may seem plausible, but the details acquired from the database may be invalid because a global dictionary or dictionary pair that accurately represents all image patches is currently unavailable.

Recently, the research focus has moved towards single image example-based SR reconstruction, which can overcome the limitations of example-based SR reconstruction using a training set. It uses only the LR image itself and can be implemented with low computational cost and memory consumption compared to example-based SR reconstruction using a training set. The fundamental assumption of this method is that a large number of similar patches may exist in the same LR image at both the same scale and across different scales. Glasner *et al.* [19] combined traditional multiple-image with example-based SR techniques and proposed the utilization of patch redundancy within and across various patch scales to reconstruct the unknown HR image. The basic prior used in [19] is that numerous similar image patches exist in most natural images. These similar patches with the same scale are regarded as patches from different LR images, whereas those with different scales are considered as HR and LR image pairs. In [20], the LR image is used to form two codebooks, which correspond to the low-frequency and high-frequency components of image patches, and by searching for the most similar image patches in the low-frequency codebook, the corresponding patch in the high-frequency codebook can be added to the target image patch to rectify the loss of high-frequency information. Chen *et al.* [21] proposed an SR algorithm that exploits the self-similarities of image patches within a single image by using the multihypothesis (MH) prediction strategy, which was also developed for compressive sensing (CS) [22] image reconstruction [23], [24] and hyperspectral image classification [25], [26]. Specifically, each patch of the input LR image is represented by a linear combination of the spatially surrounding hypothesis patches. Dong *et al.* [27] exploited the image nonlocal self-similarity to obtain good estimates of the sparse coding coefficients of the original image, before centralizing the sparse coding coefficients of the observed image relative to these estimates. Pan *et al.* [28] explored the structural self-similarity (SSSIM) index, which reflects the degree of image self-similarity, rather than comparing the reference and original images. The SSIM index is higher if there are more similar patches within one remote sensing image, thereby obtaining better reconstruction results.

In this paper, we propose a novel approach for learning only one dictionary effectively from the input LR image itself without an external HR images training set. Similar image

patches present in the same image are used to solve the SR problem as extra information. In the framework of CS theory, LR image patches can be regarded as a compressive sampling version of HR image patches under the conditions of the sparsity of signals and the incoherence between the sensing and projection matrix. When the HR image patches can be represented as a sparse linear combination of elements from a dictionary that is not coherent with the sensing matrix, we can recover the SR image patches accurately from the LR version. The CS framework is used to add this extra information to a reconstructed HR image. Obtaining extra information from similar patches in a single image is equivalent to deriving this information using different LR images, and this concept is illustrated in [28]. We use an interpolation of the LR image as the training sample in the dictionary learning stage. Our method can introduce extra information into the dictionary and allow an HR image to be obtained from a single LR image because similar image patches in the LR image provide the necessary extra information. This information can be introduced into the dictionary and used to reconstruct the HR image. The image patches can find more sparse representations [29] using our unique dictionary, which is a more compact representation of patch pairs, thereby improving the SR reconstruction results. Compared with previous methods, our algorithm can generate HR images that are competitive in quality, but our method has significantly lower computational complexity.

The remainder of this paper is organized as follows. In Section II, we explain the process employed by the training dictionary and the fast SR reconstruction algorithm based on self-example learning and sparse representation. In Section III, we report experimental evaluations of the performance of our method, which we compare with other related algorithms. Finally, we conclude this paper in Section IV.

II. FAST SR FROM SELF-EXAMPLE AND SPARSE REPRESENTATION

Theoretical results obtained from CS suggest that the sparse representation can be recovered correctly from the LR observation in mild conditions [30]. Inspired by this finding, we search for a sparse representation of each patch in the LR input and then use the coefficients of this representation to generate the HR output. In this section, we first address the sparse representation of image patches under the framework of CS, before introducing the self-example dictionary learning strategy and its approximate realization based on the K-singular value decomposition (K-SVD) algorithm [31]. Finally, the new fast single image SR algorithm based on self-example learning and sparse representation is described in detail.

A. Sparse Representation of Image Patches

An LR image patch of size $N \times N$ is converted into an $N^2 \times 1$ column vector $I_{LR} \in \mathbb{R}^p$, $p = N^2$. We want to reconstruct its corresponding $mN \times mN$ HR version, where m is a magnification factor, before converting it into an $m^2 N^2 \times 1$ column vector $I_{HR} \in \mathbb{R}^q$, $q = m^2 N^2$. The relationship

between the two column vectors can be described by a sensing matrix S :

$$I_{LR} = HL \cdot I_{HR} = S \cdot I_{HR}, \quad S \in \mathbb{R}^{p \times q}, \quad (1)$$

where H represents a blurring filter and L is the downsampling operator. If we regard I_{LR} as a degraded version of I_{HR} , then obtaining I_{HR} from the observed I_{LR} is obviously ill-posed, where the solution is not unique. However, CS theory proves that reconstructing I_{HR} is feasible under the condition of a sparse prior model. Thus, I_{HR} can be represented as a sparse linear combination of elements from a dictionary D_h , which are not coherent with the sensing matrix S :

$$I_{LR} = S \cdot I_{HR} = S \cdot D_h \alpha, \quad (2)$$

where $\alpha \in \mathbb{R}^Z$, $\|\alpha\|_0 = K \ll Z$, and Z is the number of atoms in the dictionary. Under the condition of sparsity, the HR image patch I_{HR} can be reconstructed by solving the following optimization problem:

$$\begin{cases} \min \|\alpha\|_0 \\ \text{s.t. } I_{LR} = S \cdot I_{HR} = S \cdot D_h \alpha = D_l \alpha. \end{cases} \quad (3)$$

Searching for a suitable dictionary to sparsely represent all of the HR image patches I_{HR} is a very difficult problem. In order to determine D_h , we use the bicubic interpolation version of the LR input image, which we denote as the pseudo-HR image, where it is magnified by the same magnification factor of m that is used for dictionary learning. We only use the LR input image itself to learn the dictionary without an external HR images training set; thus, the dictionary learning process is based on a self-example patch-based learning strategy. The pseudo-HR image that contains many similar image patches, as mentioned in [28], can be viewed as a corrupted version of the HR image; thus, an HR image can be reconstructed from it because similar image patches in the LR image provide the necessary extra information that can be introduced into our dictionary.

B. Selective and Efficient Sampling

Example-based SR algorithms usually involve searching patches in a large database or the input image, which are computationally expensive. The best way of reducing the SR time for real-time applications but without noticeably tampering with the quality of image is an interesting current research focus of single image example-based SR. Thus, we propose a self-example sampling method within the pseudo-HR version of input LR image. Instead of searching for patches using a similarity measure such as the Euclidean distance, we select the patch according to the threshold of the sample mean square error (SMSE) [32], [33], which reduces the computational complexity significantly.

Natural images tend to contain repetitive visual content. In particular, small (e.g., 2×2) image patches in a natural image tend to recur redundantly many times inside the image at the same scale and across different scales. This observation forms the basis of our single image SR framework without any additional external information or prior examples. When much smaller image patches are used, these similar patch repetitions

are frequent within and across image scales, even when we do not visually perceive any obvious repetitive structure in the image. This is because very small patches often contain only an edge, a corner, etc. These patches are frequent at multiple image scales in almost any natural image. Therefore, our aim is to select more representative image patches.

First, we magnify the LR input image using bicubic interpolation by the same scale factor m to produce the pseudo-HR image, where the pseudo-HR is the equivalent size to the HR that we require ultimately. Next, we sample a certain number n of different patches of size $mN \times mN$, which are distributed throughout the entire pseudo-HR image X_m . The n patches are selected randomly to obtain sufficient patches for HR image reconstruction. This ensures that we collect a large number of similar image patch pairs from the LR image. We convert each patch into an $m^2 N^2 \times 1$ column vector one by one as $X_n = \{x_1, x_2, \dots, x_{n-1}, x_n\}$, $x_i \in \mathbb{R}^q$, $i = 1, 2, \dots, n - 1, n$ and we compute the SMSE within each x_i

$$\text{SMSE}(x_i) = \frac{\sum_{j=1}^{m^2 N^2} (x_{ij} - \sum_{j=1}^{m^2 N^2} x_{ij}/m^2 N^2)^2}{m^2 N^2 - 1}, \quad (4)$$

where x_i represents the column vector in X_n and x_{ij} is the element of the j -th row in x_i , $j = 1, 2, \dots, m^2 N^2 - 1, m^2 N^2$. To find the image patches that can sparsely represent the HR image patches to ensure that the reconstruction error is minimized, a threshold λ is used to remove all the smooth column vectors with SMSEs that are less than λ . This is because there are many similar image patches in a pseudo-HR image but only some of them contain the necessary HR or detail information required to reconstruct HR image patches. To identify the patches that contain HR or detail information from the total n image patches, we propose a fast and efficient sampling strategy based on SMSE. In terms of theory, the SMSE is higher when more HR and detail information is contained in one image patch. In addition, based on a large number of experiments, we found that the effect of HR image patch reconstruction using smooth pseudo-HR image patches was not as obvious as we expected and it required a large amount of computation time. Based on this theoretical observation, we set the SMSE threshold to extract useful image patches. All of the similar patches are retained for use as necessary extra information, which can be introduced into the dictionary employed to reconstruct the HR image. Finally, we denote all the remaining pseudo-HR image patch column vectors in the set as $X_h = \{x_1, x_2, \dots, x_{Q-1}, x_Q\}$, $x_i \in \mathbb{R}^q$, $i = 1, 2, \dots, Q - 1, Q$.

This sampling strategy based on self-example considers both the global information and the specific details. For example, if we sample $n = 3000$ patches of size $mN \times mN$ from pseudo-HR image X_m , there are 3000 atoms in X_n . After filtering by SMSE, the number of atoms in the column vectors set X_h is ten times lower than that in X_n on average, which means that there are around 300 atoms in X_h . SMSE can ensure that all the necessary extra HR and detail information are retained, and it also can simplify much of the subsequent work without increasing the computational cost and it facilitates dimensionality reduction for our dictionary D_h , thereby leading to a fast reconstruction phase. During the HR patch recon-

struction procedures for a specific image in most algorithms, the quantity of patches sampled starting from one patch and moving to another with overlapping pixels in each direction tends to be similar or even equivalent. The intrinsic factor that affects the reconstruction speed is the size N of sampled patches and dimensionality Z of atoms in the dictionary, whereas the main extrinsic factor is the complexity of different reconstruction algorithms. The dimensionality Z of atoms in the dictionary immediately affects the efficiency of sparse coding during dictionary preparation and the reconstruction of sparse coefficients. The experimental results in Section III show that the proposed algorithm is much faster than several state-of-the-art approaches and it can achieve comparable or even superior quality. Consequently, we propose a much more accurate model for sparse coding by exploiting the local and nonlocal redundancies within input images.

C. Self-Example Dictionary Preparation

Recent research [34] into image statistics has shown that image patches can be represented well as sparse linear combination of elements from an appropriately selected over-complete dictionary. Inspired by this observation, a self-example patch-based dictionary learning strategy is proposed in this section and a K-SVD algorithm is employed to learn this dictionary to allow the sparse representation of LR input image patches.

Assume that the column vectors in X_h can be represented by a sparse linear combination in a dictionary $D_h = [d_1, d_2, \dots, d_{Z-1}, d_Z] \in \mathbb{R}^{q \times Z}$ with each $\|d_j\|_2 = 1$, $j = 1, 2, \dots, Z - 1, Z$:

$$x_i = D_h \cdot \beta_i, \quad \beta_i \in \mathbb{R}^Z, \quad \|\beta_i\|_0 \ll Z \quad (5)$$

Denote a sparse coefficient matrix $\beta = [\beta_1, \beta_2, \dots, \beta_{Q-1}, \beta_Q]$ and we aim to design a dictionary that minimizes the reconstruction error for a set of image patches.

$$\begin{cases} \min_{D_h, \beta} \|X_h - D_h \beta\|_F^2 \\ \text{s.t. } \|\beta_i\|_0 \leq Z, \quad i = 1, 2, \dots, Q - 1, Q \end{cases} \quad (6)$$

The K-SVD dictionary learning algorithm is used to solve this problem based on a self-example patch-based learning strategy, which can be described by two essential procedures: 1) search for the sparse coding of X_h when the dictionary is fixed; and 2) update each atom (each column in the dictionary D_h) and their corresponding coefficients (each row in the sparse coefficient matrix β). K-SVD is an iterative algorithm that updates the signal sparse coding in the current dictionary and dictionary atoms alternately in order to map the current signal preferentially. Overall, updating the dictionary atoms and their corresponding sparse coefficients is a system that accelerates the convergence speed of our algorithm greatly.

More specifically, the sparse coding problem can be implemented using any pursuit algorithms such as basis pursuit (BP) and orthogonal matching pursuit (OMP) [30]. In this paper, we use OMP to minimize the objective function in (6), which is much faster than ℓ^1 -optimization-based methods [17]. Moreover, Batch-OMP [35] is an implementation of the OMP algorithm that is optimized specifically for sparse coding

large sets of signals using the same dictionary. However, the magnitude of our signals X_h , which are only sampled from pseudo-HR without a substantial external training set and they have already been filtered previously, is much lower than that in [17], [35]. Based on complexity analysis and the total number of operations required to sparse code a variable number of signals using both OMP and Batch-OMP in [35], we employ straightforward OMP, which is faster and more suitable for sparse coding a relatively small amount of signals with K-SVD.

In the dictionary learning process, we update the dictionary by columns, where one atom in the dictionary is updated each time and the other atoms in the dictionary remain fixed. We denote the k^{th} atom of the dictionary D_h as d_k and the k^{th} row in the sparse coefficients matrix β as β^k . Then, we transform (6) such that $d_k \beta^k$ can be separated from $D_h \beta$ [31], as follows

$$\begin{aligned} \|X_h - D_h \beta\|_F^2 &= \|X_h - \sum_{j=1}^Z d_j \beta^j\|_F^2 \\ &= \|E_k - d_k \beta^k\|_F^2, \end{aligned} \quad (7)$$

where $E_k = X_h - \sum_{j \neq k} d_j \beta^j$ stands for the error for all Q examples when the k^{th} atom is removed. Before this problem can be solved directly by SVD, β^k is very likely to be filled, since we do not preserve the constraint in (6). To achieve this, the update step uses only the signals in X_h whose sparse representations use the current atom. Let Ψ denote the indices of the signals in X_h which include the k^{th} atom. $E_k^\Psi = X_h^\Psi - \sum_{j \neq k} d_j \beta_{\Psi}^j$ is the error matrix without the k^{th} atom, d_k is the updated atom, and β_Ψ^k is the new row of coefficients in β_Ψ .

The entire K-SVD algorithm only converges to a local minimum instead of a global one, and the analysis provided in [31] only assumes a reduction of the target function value in (7), thereby improving a given initial dictionary D_h rather than finding an optimal solution. Therefore, a much faster approach is to utilize an approximation of (7) as long as it ensures a reduction of the final target function.

In our implementation, we replace the exact SVD computation with a much faster approximation, which employs a single iteration of alternate-optimization over the atom d_k and the coefficients row β^k , which is given as follows

$$\begin{aligned} d_k &= E_k (\beta^k)^T / \|E_k (\beta^k)^T\|_2 \\ \beta^k &= (d_k)^T E_k, \end{aligned} \quad (8)$$

then we replace the non- Ψ indices (original zero entries) in β^k with zeros to preserve the constraint in (6).

This process is known to ultimately converge to the optimum, and supplies an approximation which reduces the penalty term. Our experiment shows that a single iteration of this process is sufficient to provide very close results to the full computation. A significant advantage of this approximation is that it eliminates the need to explicitly decompose the matrix E_k^Ψ . The exact SVD computation is both time and memory consuming. It can be avoided by only computing the products of the matrix E_k with vectors in the approximate formula (8).

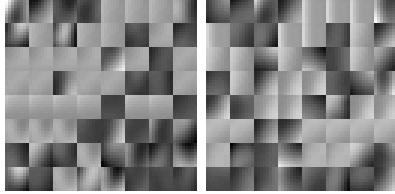


Fig. 1. Two HR image patch dictionaries trained by Eq. 6 using $n = 3000$ pseudo-HR image patch pairs sampled from the bicubic interpolation versions of LR images (128×128) of Lena and Barbara magnified by a factor of $m = 4$. A dictionary of size 64×64 was learned with an SMSE threshold of $\lambda = 1100$ for each pseudo-HR image patch of size 8×8 . Left to right: dictionary for Lena and dictionary for Barbara.

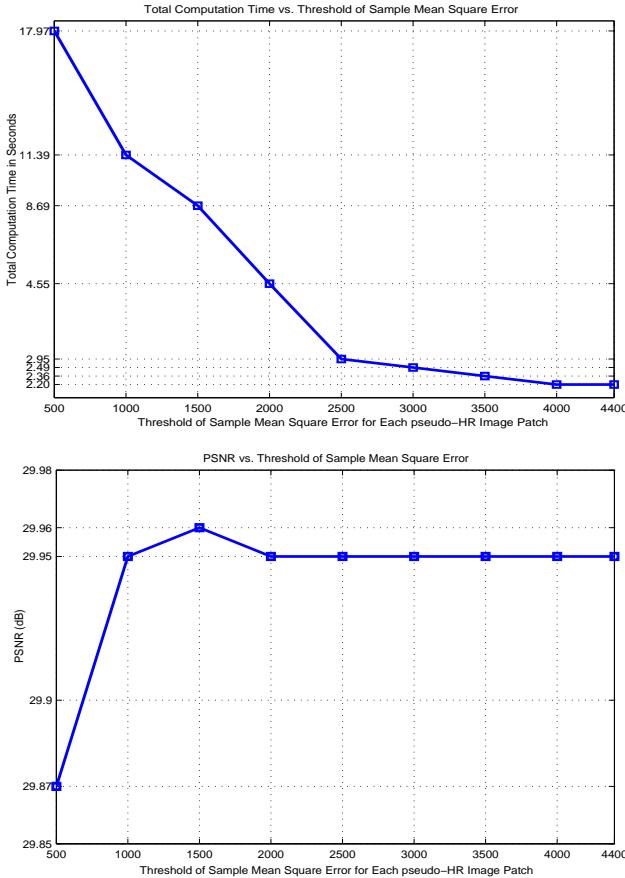


Fig. 2. The total computation time (including the dictionary learning and reconstruction phases) and PSNR of the reconstructed “Clown” image (256×256) magnified by a factor of $m = 2$ with various sample mean square error thresholds for each pseudo-HR image patch (in seconds and dB, respectively).

D. Fast SR

Given a single input LR image Y of size $B \times B$ and a magnification factor m , we want to reconstruct the HR image X of size $mB \times mB$. In the first step, we obtain the pseudo-HR image X_m of size $mB \times mB$. Y is then partitioned into nonoverlapping patches of size $N \times N$. For each LR patch y , a corresponding $mN \times mN$ pseudo-HR patch exists in X_m at the same spatial location as y in Y .

Then, we exploit the sampling strategy discussed in Section II-B and obtain the image patch column vectors set X_h based on the SMSE threshold λ . Simultaneously, we employ the approximate K-SVD algorithm to train the dictionary D_h and

we use the OMP algorithm to obtain the sparse coefficients β with X_h in (6). In dictionary update stage, we calculate d_k and β^k using (8) and ensure that the reconstruction error is minimized until (7) remains unchanged.

In the image patch reconstruction procedure, we start from the upper-left corner where o pixels overlap in each direction and we also use the OMP algorithm to solve the corresponding sparse coefficient α_i for each LR patch I_{LR}^i , $i = 1, 2, \dots, T - 1$, where $T = (B/N)^2$:

$$\min_{\alpha_i} \|I_{LR}^i - S \cdot D_h \alpha_i\|_2^2 \text{ s.t. } \|\alpha_i\|_0 \leq Z. \quad (9)$$

In general, the optimization problem of (9) is NP-hard but recent results [36] suggest that the desired coefficients α_i can be recovered efficiently by minimizing the ℓ^1 -norm provided that they are sufficiently sparse. However, we still use the faster ℓ^0 -norm for matching the algorithm, which is used as the sparse coding algorithm in approximate K-SVD, because of the better learning and reconstruction effects with the same criterion. We can then generate each corresponding HR image patch after finding the sparse representations, as follows.

$$\hat{I}_{HR} = D_h \alpha_i \quad (10)$$

The reconstruction obtained from (10) yields an estimate \hat{I}_{HR} of the HR image \hat{X} . To enforce a global reconstruction constraint, we also project this initial reconstructed HR image onto the solution space of $Y = S \cdot X$:

$$X^* = \arg \min_X \|S \cdot X - Y\|_2^2 + c\|X - \hat{X}\|_2^2. \quad (11)$$

This optimization problem can be computed efficiently using gradient descent. The update equation for this iterative method is

$$X_{t+1} = X_t + v[S^T \cdot (Y - S \cdot X_t) + c(X - \hat{X})], \quad (12)$$

where X_t is the estimate of the HR image after the t -th iteration and v is the step size of the gradient descent. We use result X^* obtained from this optimization as our final estimate of the HR image. This image is as close as possible to the initial reconstructed \hat{X} given by sparsity, while it respects the reconstruction constraint.

The procedure of the SR reconstruction algorithm based on the self-example patch-based dictionary machine learning strategy and sparse representation in the framework of CS theory is summarized by Algorithm 1. We consider that the sensing matrix S has a bicubic interpolation relationship with dictionary D_h and D_l in (3) because the pseudo-HR image is obtained by bicubic interpolation from the LR image, which means that D_h can be obtained by bicubic interpolation from D_l .

III. EXPERIMENTAL RESULTS

In our experiments, we magnified 13 LR grayscale natural images using bicubic interpolation, Yang *et al.*'s method, Zeyde *et al.*'s method, Chen *et al.*'s method, Dong *et al.*'s method, and our proposed approach with a factor of $m = 2$ and $m = 4$ by employing the processes described in the previous sections, and we investigated the performance of our

TABLE I

SUMMARY OF PSNR (DB), RMSE, SSIM, AND RECONSTRUCTION TIME (S) RESULTS FOR 13 TEST IMAGES, WHERE SIZE = 128×128 , SCALE FACTOR $m = 2$, THRESHOLD OF SMSE $\lambda = 900$, AND PATCH NUMBER $n = 3000$ (TOTAL COMPUTATION TIME IN PARENTHESES USING THE PROPOSED METHOD).

$m = 2, \lambda = 900, n = 3000$									
Algorithm	PSNR	RMSE	SSIM	TIME	Algorithm	PSNR	RMSE	SSIM	TIME
Lena					Crowd				
Bicubic	31.09	7.11	0.9249	0.10	Bicubic	28.52	9.56	0.9008	0.11
Yang <i>et al.</i>	33.48	5.40	0.9513	114.46	Yang <i>et al.</i>	30.59	7.54	0.9378	123.60
Zeyde <i>et al.</i>	32.89	5.78	0.9502	2.42	Zeyde <i>et al.</i>	30.18	7.90	0.9315	2.44
Chen <i>et al.</i>	32.71	5.90	0.9491	184.69	Chen <i>et al.</i>	29.84	8.22	0.9300	159.87
Dong <i>et al.</i>	33.42	5.44	0.9484	224.06	Dong <i>et al.</i>	30.55	7.57	0.9335	233.51
Proposed	32.45	6.01	0.9458	1.71 (3.59)	Proposed	29.92	8.14	0.9286	1.72 (9.30)
Barbara					Girl				
Bicubic	29.48	8.56	0.8688	0.11	Bicubic	32.66	5.93	0.9062	0.10
Yang <i>et al.</i>	30.93	7.25	0.8961	113.84	Yang <i>et al.</i>	34.98	4.55	0.9460	120.52
Zeyde <i>et al.</i>	30.76	7.39	0.8922	2.45	Zeyde <i>et al.</i>	34.97	4.55	0.9469	2.44
Chen <i>et al.</i>	30.52	7.60	0.8917	173.38	Chen <i>et al.</i>	34.69	4.69	0.9404	159.90
Dong <i>et al.</i>	30.97	7.21	0.8945	233.84	Dong <i>et al.</i>	35.00	4.53	0.9472	235.25
Proposed	30.45	7.61	0.8913	1.76 (7.79)	Proposed	34.55	4.74	0.9402	1.69 (2.91)
Barbara2					Gold Hill				
Bicubic	28.56	9.52	0.8218	0.12	Bicubic	30.79	7.36	0.8720	0.09
Yang <i>et al.</i>	29.66	8.38	0.8678	122.63	Yang <i>et al.</i>	32.11	6.33	0.9063	123.08
Zeyde <i>et al.</i>	29.59	8.45	0.8663	2.45	Zeyde <i>et al.</i>	31.89	6.49	0.9017	2.44
Chen <i>et al.</i>	29.30	8.73	0.8624	161.12	Chen <i>et al.</i>	31.61	6.70	0.8999	174.35
Dong <i>et al.</i>	29.70	8.35	0.8690	237.34	Dong <i>et al.</i>	32.26	6.21	0.9050	233.51
Proposed	29.49	8.55	0.8657	1.78 (2.77)	Proposed	31.72	6.61	0.9033	1.72 (2.28)
Boat					Man				
Bicubic	28.27	9.84	0.8615	0.12	Bicubic	29.73	8.31	0.8789	0.11
Yang <i>et al.</i>	30.01	8.06	0.9048	122.46	Yang <i>et al.</i>	31.76	6.58	0.9197	123.79
Zeyde <i>et al.</i>	29.59	8.45	0.8991	2.44	Zeyde <i>et al.</i>	31.42	6.85	0.9193	2.44
Chen <i>et al.</i>	29.35	8.66	0.8965	160.79	Chen <i>et al.</i>	31.05	7.14	0.9112	160.16
Dong <i>et al.</i>	30.01	8.05	0.9040	234.35	Dong <i>et al.</i>	31.69	6.64	0.9145	235.92
Proposed	29.32	8.70	0.8952	1.71 (2.34)	Proposed	30.90	7.23	0.9103	1.70 (2.63)
Cameraman					Peppers				
Bicubic	28.87	9.19	0.9157	0.10	Bicubic	31.03	7.16	0.9462	0.10
Yang <i>et al.</i>	31.32	6.93	0.9463	123.78	Yang <i>et al.</i>	33.57	5.34	0.9663	117.11
Zeyde <i>et al.</i>	30.88	7.29	0.9431	2.44	Zeyde <i>et al.</i>	33.36	5.48	0.9627	2.44
Chen <i>et al.</i>	30.38	7.64	0.9412	161.80	Chen <i>et al.</i>	32.41	6.11	0.9604	159.87
Dong <i>et al.</i>	31.40	6.86	0.9487	221.84	Dong <i>et al.</i>	33.65	5.30	0.9658	226.46
Proposed	30.30	7.78	0.9408	1.73 (9.18)	Proposed	32.46	6.08	0.9607	1.78 (9.71)
Clown					Mandrill				
Bicubic	28.72	9.34	0.8834	0.11	Bicubic	25.67	13.28	0.7385	0.10
Yang <i>et al.</i>	30.41	7.69	0.9156	113.42	Yang <i>et al.</i>	26.42	12.18	0.8038	129.30
Zeyde <i>et al.</i>	30.46	7.65	0.9180	2.43	Zeyde <i>et al.</i>	26.40	12.21	0.8055	2.43
Chen <i>et al.</i>	29.85	8.23	0.9081	169.59	Chen <i>et al.</i>	26.28	12.37	0.7988	178.32
Dong <i>et al.</i>	30.49	7.62	0.9140	226.65	Dong <i>et al.</i>	26.38	12.23	0.7933	239.40
Proposed	29.94	8.12	0.9097	1.81 (11.59)	Proposed	26.39	12.23	0.8034	1.66 (2.16)
Couple					Average				
Bicubic	28.30	9.80	0.8426	0.10	Bicubic	29.36	8.84	0.8739	0.11
Yang <i>et al.</i>	29.71	8.34	0.8861	115.99	Yang <i>et al.</i>	31.15	7.27	0.9114	120.31
Zeyde <i>et al.</i>	29.40	8.64	0.8812	2.44	Zeyde <i>et al.</i>	30.91	7.47	0.9091	2.44
Chen <i>et al.</i>	29.25	8.79	0.8793	160.35	Chen <i>et al.</i>	30.55	7.76	0.9053	166.48
Dong <i>et al.</i>	29.79	8.26	0.8864	236.08	Dong <i>et al.</i>	31.18	7.25	0.9096	232.17
Proposed	29.37	8.67	0.8795	1.75 (7.15)	Proposed	30.57	7.72	0.9057	1.73 (5.64)

proposed method. The peak signal-to-noise ratio (PSNR), root mean squared error (RMSE), structural similarity (SSIM) [37], and elapsed time were used as quality measures.

For Yang *et al.*'s sparse representation SR [16], which employs ℓ^1 -optimization-based methods with simultaneous dictionary pair learning and reconstruction phases, LR patches of size 5×5 with an overlap of 4 pixels between adjacent patches and a sparsity regularization coefficient of 0.1 were used in all of the experiments, as suggested in [11]. The implementation¹ was obtained from the authors and it was used with their pretrained 25×1024 dictionary for D_h and a 100×1024 dictionary for D_l . In particular, the dictionary

pair learning process required about 22 hours to obtain 50000 training patch-pairs from a large training dataset that contained hundreds of images.

Compared to Yang *et al.*'s algorithms, Zeyde *et al.*'s implementation² [17] uses the same idea of a training phase and a reconstruction phase, with sparse modeling of the desired image patches, and a pair of dictionaries is used to migrate from the LR domain to the HR, but different algorithms are used for dictionary pair learning from the same training dataset: K-SVD for the LR dictionary D_l and pseudo-inverse for the HR dictionary D_h . It should be noted that Zeyde *et al.*'s implementation packages (K-SVD and OMP) use the MEX

¹<http://www.ifp.illinois.edu/~jyang29/ScSR.htm>

²http://www.cs.technion.ac.il/~elad/Various/Single_Image_SR.zip

TABLE II

SUMMARY OF PSNR (DB), RMSE, SSIM, AND RECONSTRUCTION TIME (S) RESULTS FOR 13 TEST IMAGES, WHERE SIZE = 256×256 , SCALE FACTOR $m = 2$, THRESHOLD OF SMSE $\lambda = 1000$, AND PATCH NUMBER $n = 3000$ (TOTAL COMPUTATION TIME IN PARENTHESES USING THE PROPOSED METHOD).

$m = 2, \lambda = 1000, n = 3000$									
Algorithm	PSNR	RMSE	SSIM	TIME	Algorithm	PSNR	RMSE	SSIM	TIME
Lena					Crowd				
Bicubic	34.12	5.02	0.9905	0.09	Bicubic	32.66	5.93	0.9919	0.10
Yang <i>et al.</i>	36.11	3.99	0.9967	499.24	Yang <i>et al.</i>	35.14	4.46	0.9983	474.36
Zeyde <i>et al.</i>	35.79	4.14	0.9971	7.11	Zeyde <i>et al.</i>	34.75	4.16	0.9987	7.18
Chen <i>et al.</i>	35.58	4.24	0.9978	491.77	Chen <i>et al.</i>	34.08	5.12	0.9974	400.90
Dong <i>et al.</i>	36.21	3.94	0.9909	1072.01	Dong <i>et al.</i>	35.20	4.43	0.9940	1206.62
Proposed	35.81	4.15	0.9979	6.50 (9.59)	Proposed	34.27	4.93	0.9987	6.51 (12.71)
Barbara					Girl				
Bicubic	25.35	13.77	0.9461	0.10	Bicubic	36.64	3.76	0.9918	0.13
Yang <i>et al.</i>	25.92	12.89	0.9809	476.72	Yang <i>et al.</i>	38.68	2.97	0.9976	497.15
Zeyde <i>et al.</i>	25.80	13.08	0.9809	7.06	Zeyde <i>et al.</i>	38.56	3.01	0.9984	7.10
Chen <i>et al.</i>	25.78	13.10	0.9815	327.74	Chen <i>et al.</i>	37.96	3.22	0.9982	338.01
Dong <i>et al.</i>	25.54	13.48	0.9187	1114.39	Dong <i>et al.</i>	38.89	2.90	0.9946	1168.30
Proposed	25.87	12.97	0.9819	6.49 (8.02)	Proposed	38.05	3.02	0.9984	6.50 (11.09)
Barbara2					Gold Hill				
Bicubic	27.78	10.41	0.9684	0.09	Bicubic	31.45	6.82	0.9826	0.12
Yang <i>et al.</i>	28.96	9.09	0.9941	514.42	Yang <i>et al.</i>	32.54	6.02	0.9947	517.67
Zeyde <i>et al.</i>	28.72	9.34	0.9946	7.08	Zeyde <i>et al.</i>	32.36	6.15	0.9953	7.11
Chen <i>et al.</i>	28.67	9.40	0.9946	342.80	Chen <i>et al.</i>	32.23	6.24	0.9951	350.64
Dong <i>et al.</i>	29.05	9.00	0.9664	1186.79	Dong <i>et al.</i>	32.59	5.98	0.9823	1172.78
Proposed	28.77	9.29	0.9947	6.50 (7.46)	Proposed	32.34	6.13	0.9953	6.50 (9.53)
Boat					Man				
Bicubic	29.95	8.11	0.9834	0.10	Bicubic	31.05	7.15	0.9848	0.09
Yang <i>et al.</i>	31.50	6.78	0.9950	516.09	Yang <i>et al.</i>	32.79	5.85	0.9961	517.41
Zeyde <i>et al.</i>	31.16	7.06	0.9955	7.09	Zeyde <i>et al.</i>	32.48	6.06	0.9967	7.14
Chen <i>et al.</i>	30.97	7.21	0.9955	331.50	Chen <i>et al.</i>	32.16	6.29	0.9965	329.08
Dong <i>et al.</i>	31.58	6.72	0.9854	1166.43	Dong <i>et al.</i>	32.75	5.88	0.9858	1171.04
Proposed	31.15	7.06	0.9956	6.52 (7.41)	Proposed	32.40	6.03	0.9967	6.50 (8.16)
Cameraman					Peppers				
Bicubic	35.73	4.17	0.9939	0.11	Bicubic	31.76	6.59	0.9917	0.08
Yang <i>et al.</i>	38.69	2.96	0.9978	523.61	Yang <i>et al.</i>	32.89	5.78	0.9965	520.85
Zeyde <i>et al.</i>	38.68	2.97	0.9994	7.10	Zeyde <i>et al.</i>	32.70	5.91	0.9975	7.14
Chen <i>et al.</i>	37.96	3.22	0.9993	307.23	Chen <i>et al.</i>	32.35	6.15	0.9974	334.96
Dong <i>et al.</i>	39.38	2.74	0.9969	1193.69	Dong <i>et al.</i>	32.98	5.72	0.9913	1232.00
Proposed	38.36	3.03	0.9995	6.51 (12.21)	Proposed	32.66	5.94	0.9975	6.48 (10.26)
Clown					Mandrill				
Bicubic	32.65	5.95	0.9880	0.10	Bicubic	23.62	16.79	0.9550	0.14
Yang <i>et al.</i>	34.82	4.63	0.9952	476.62	Yang <i>et al.</i>	24.39	15.39	0.9888	528.07
Zeyde <i>et al.</i>	34.77	4.66	0.9959	7.03	Zeyde <i>et al.</i>	24.39	15.39	0.9888	7.15
Chen <i>et al.</i>	33.81	5.10	0.9914	405.01	Chen <i>et al.</i>	24.26	15.61	0.9889	331.72
Dong <i>et al.</i>	34.93	4.58	0.9911	1283.60	Dong <i>et al.</i>	24.31	15.52	0.9496	1262.22
Proposed	34.35	4.79	0.9961	6.55 (12.97)	Proposed	24.32	15.49	0.9891	6.51 (7.12)
Couple					Average				
Bicubic	29.55	8.49	0.9808	0.09	Bicubic	30.95	7.92	0.9807	0.10
Yang <i>et al.</i>	30.96	7.22	0.9948	501.43	Yang <i>et al.</i>	32.57	6.77	0.9943	504.90
Zeyde <i>et al.</i>	30.75	7.40	0.9953	7.14	Zeyde <i>et al.</i>	32.38	6.87	0.9949	7.11
Chen <i>et al.</i>	30.40	7.69	0.9949	398.40	Chen <i>et al.</i>	32.02	7.12	0.9945	360.75
Dong <i>et al.</i>	31.01	7.18	0.9807	1242.29	Dong <i>et al.</i>	32.65	6.77	0.9791	1190.17
Proposed	30.52	7.59	0.9953	6.51 (11.38)	Proposed	32.22	6.96	0.9951	6.51 (9.84)

function to call a C program in MATLAB whereas all of the other methods are implemented solely in MATLAB. Around 100000 training patch-pairs were collected and dictionary pair learning required approximately 10 minutes to complete 40 iterations of the K-SVD algorithm with 1000 atoms in the dictionary. We used 3×3 for LR patches with an overlap of 2 pixels between adjacent patches in our comparison experiments. If Zeyde *et al.*'s method is implemented purely in MATLAB, such as magnifying an LR image of size 128×128 by a factor of $m = 4$, the time required for the dictionary pair learning and reconstruction phases will increase considerably to an average of about 1 hour and 5 minutes for each image.

In Chen *et al.*'s SR method³ [21] using MH prediction, the pseudo-HR image is magnified by bicubic interpolation from the LR image. We magnified the LR image by a factor of $m = 2$ with LR patches fixed to a size of 4×4 in Y and the search window size was fixed to 8 for hypothesis generation. To magnify the LR image by a factor of $m = 4$, we first magnified the LR image by a factor of $m = 2$ and then magnified the resulting image again to obtain the final result. To facilitate rapid reconstruction with both $m = 2$ and $m = 4$, we only used one iteration in each $m = 2$ magnification process during our experiments.

³<https://sites.google.com/site/chenresearchsite/publications>

TABLE III

SUMMARY OF PSNR (DB), RMSE, SSIM, AND RECONSTRUCTION TIME (S) RESULTS FOR 13 TEST IMAGES, WHERE SIZE = 128×128 , SCALE FACTOR $m = 4$, THRESHOLD OF SMSE $\lambda = 1500$, AND PATCH NUMBER $n = 3000$ (TOTAL COMPUTATION TIME IN PARENTHESES USING THE PROPOSED METHOD).

$m = 4, \lambda = 1500, n = 3000$									
Algorithm	PSNR	RMSE	SSIM	TIME	Algorithm	PSNR	RMSE	SSIM	TIME
Lena					Crowd				
Bicubic	28.77	9.29	0.9142	0.11	Bicubic	26.51	12.05	0.8906	0.10
Yang <i>et al.</i>	29.53	8.51	0.9334	455.87	Yang <i>et al.</i>	27.34	10.96	0.9175	488.64
Zeyde <i>et al.</i>	29.57	8.47	0.9351	2.81	Zeyde <i>et al.</i>	27.42	10.85	0.9189	2.81
Chen <i>et al.</i>	29.57	8.47	0.9351	387.05	Chen <i>et al.</i>	27.23	11.08	0.9169	405.70
Dong <i>et al.</i>	29.60	8.44	0.9364	997.46	Dong <i>et al.</i>	27.40	10.88	0.9230	1072.04
Proposed	29.58	8.46	0.9352	2.98 (6.77)	Proposed	27.43	10.84	0.9294	2.97 (5.77)
Barbara					Girl				
Bicubic	23.58	16.88	0.8121	0.12	Bicubic	30.68	7.46	0.8962	0.11
Yang <i>et al.</i>	23.81	16.44	0.8313	464.13	Yang <i>et al.</i>	31.74	6.60	0.9263	492.42
Zeyde <i>et al.</i>	23.84	16.39	0.8320	2.80	Zeyde <i>et al.</i>	31.81	6.55	0.9265	2.83
Chen <i>et al.</i>	23.82	16.43	0.8320	405.19	Chen <i>et al.</i>	31.80	6.55	0.9268	402.90
Dong <i>et al.</i>	23.84	16.39	0.8321	1023.14	Dong <i>et al.</i>	31.81	6.55	0.9261	1012.33
Proposed	23.85	16.38	0.8321	2.89 (5.72)	Proposed	31.83	6.53	0.9383	2.99 (6.21)
Barbara2					Gold Hill				
Bicubic	24.60	15.01	0.7860	0.10	Bicubic	27.60	10.63	0.8495	0.09
Yang <i>et al.</i>	24.96	14.41	0.8265	490.96	Yang <i>et al.</i>	28.05	10.09	0.8785	484.15
Zeyde <i>et al.</i>	24.98	14.37	0.8288	2.82	Zeyde <i>et al.</i>	28.09	10.05	0.8806	2.80
Chen <i>et al.</i>	24.87	14.55	0.8260	407.72	Chen <i>et al.</i>	27.95	10.21	0.8769	398.45
Dong <i>et al.</i>	24.99	14.36	0.8288	1018.04	Dong <i>et al.</i>	28.07	10.07	0.8789	1015.41
Proposed	25.01	14.34	0.8401	2.98 (5.69)	Proposed	28.10	10.04	0.8907	2.98 (6.32)
Boat					Man				
Bicubic	25.48	13.57	0.8407	0.10	Bicubic	26.80	11.66	0.8602	0.11
Yang <i>et al.</i>	26.01	12.77	0.8731	489.85	Yang <i>et al.</i>	27.37	10.91	0.8897	496.31
Zeyde <i>et al.</i>	26.06	12.69	0.8752	2.76	Zeyde <i>et al.</i>	27.41	10.87	0.8915	2.79
Chen <i>et al.</i>	26.01	12.77	0.8752	412.65	Chen <i>et al.</i>	27.40	10.88	0.8915	396.10
Dong <i>et al.</i>	26.07	12.68	0.8752	1004.58	Dong <i>et al.</i>	27.39	10.89	0.8915	1017.06
Proposed	26.07	12.68	0.8850	2.99 (7.20)	Proposed	27.41	10.87	0.8920	2.98 (7.36)
Cameraman					Peppers				
Bicubic	27.44	10.83	0.9096	0.11	Bicubic	27.86	10.32	0.9369	0.10
Yang <i>et al.</i>	28.50	9.58	0.9315	497.93	Yang <i>et al.</i>	28.35	9.75	0.9476	475.80
Zeyde <i>et al.</i>	28.54	9.54	0.9402	2.78	Zeyde <i>et al.</i>	28.60	9.47	0.9485	2.78
Chen <i>et al.</i>	28.50	9.58	0.9327	419.63	Chen <i>et al.</i>	28.46	9.62	0.9485	403.78
Dong <i>et al.</i>	28.54	9.54	0.9400	1002.39	Dong <i>et al.</i>	28.63	9.44	0.9497	1010.78
Proposed	28.54	9.54	0.9406	2.97 (9.22)	Proposed	28.64	9.43	0.9528	3.01 (8.13)
Clown					Mandrill				
Bicubic	26.64	11.88	0.8698	0.11	Bicubic	21.05	22.59	0.6849	0.11
Yang <i>et al.</i>	27.37	10.92	0.8914	460.75	Yang <i>et al.</i>	21.29	21.98	0.7435	514.63
Zeyde <i>et al.</i>	27.39	10.89	0.8998	2.77	Zeyde <i>et al.</i>	21.30	21.95	0.7468	2.78
Chen <i>et al.</i>	27.29	11.01	0.8891	427.75	Chen <i>et al.</i>	21.25	22.06	0.7495	401.39
Dong <i>et al.</i>	27.36	10.92	0.8911	1103.40	Dong <i>et al.</i>	21.24	22.10	0.7435	1221.07
Proposed	27.39	10.89	0.8999	3.03 (19.84)	Proposed	21.34	21.86	0.7532	2.98 (5.89)
Couple					Average				
Bicubic	25.33	13.81	0.8176	0.10	Bicubic	26.33	12.77	0.8514	0.11
Yang <i>et al.</i>	25.83	13.04	0.8521	470.02	Yang <i>et al.</i>	26.93	12.00	0.8802	483.19
Zeyde <i>et al.</i>	25.84	13.02	0.8542	2.78	Zeyde <i>et al.</i>	26.99	11.93	0.8829	2.79
Chen <i>et al.</i>	25.76	13.13	0.8526	400.56	Chen <i>et al.</i>	26.91	12.02	0.8810	405.30
Dong <i>et al.</i>	25.84	13.02	0.8531	1113.77	Dong <i>et al.</i>	26.98	11.94	0.8823	1047.04
Proposed	25.85	12.99	0.8635	2.97 (6.14)	Proposed	27.00	11.91	0.8886	2.98 (7.71)

In Dong *et al.*'s nonlocally centralized sparse representation (NCSR) model [27], the simulated LR image is generated by blurring an HR image with a 7×7 Gaussian kernel with a standard deviation of 1.6, before downsampling the blurred image by scaling factors of $m = 2$ and $m = 4$ in both the horizontal and vertical directions. NCSR clusters the patches in an LR image into K clusters and learns a sub-dictionary based on principal components analysis (PCA) of each cluster. For a given patch, NCSR first checks the cluster to which it belongs by calculating the distances to the means of the clusters, before coding it using the PCA sub-dictionary of the cluster.

In our proposed method, we first magnified the LR input image by bicubic interpolation to obtain the pseudo-HR image

X_m . We sampled $n = 3000$ image patches with LR patches of size $N = 2$ with an overlap of $o = 1$ pixel between adjacent patches in each direction using magnification factors of $m = 2$ and $m = 4$; thus, the pseudo-HR image patches were fixed to a size of 4×4 and 8×8 . All of the experiments were performed using MATLAB R2013a on an Intel (R) Core (TM) i7-2600 @ 3.40 GHz machine with 8 GB of RAM.

Figure 1 shows two HR dictionaries trained using pseudo-HR images of "Barbara" and "Lena", respectively. The learned dictionary included the basic patterns of the image patches instead of the raw patch prototypes, due to its compactness. Figure 2 shows the total computation time in seconds, including the dictionary learning and reconstruction phases, and the

TABLE IV

PSNR (dB) AND TOTAL COMPUTATION TIME (S) FOR RECONSTRUCTED IMAGES USING DIFFERENT TEST IMAGE SIZES, DIFFERENT QUANTITIES OF SAMPLED IMAGE PATCHES, AND VARIOUS MAXIMUM SMSE THRESHOLDS.

Image	$n = 3000$					$n = 5000$					$n = 10000$				
	m	size	λ	PSNR	TIME	m	size	λ	PSNR	TIME	m	size	λ	PSNR	TIME
Lena	2	128	1900	32.45	1.97	2	128	2000	32.45	1.94	2	128	2000	32.46	2.01
		256	1500	35.81	6.67		256	1800	35.80	6.59		256	1900	35.80	6.63
	4	128	1500	29.58	6.77	4	128	1700	29.59	5.96	4	128	2000	29.58	5.77
Cameraman	2	128	3800	30.30	1.98	2	128	4100	30.30	1.92	2	128	4100	30.30	2.00
		256	2600	38.36	6.76		256	2700	38.36	6.62		256	2700	38.36	6.73
	4	128	2200	28.54	5.81	4	128	2900	28.53	5.34	4	128	3500	28.53	5.42
Clown	2	128	4400	29.95	2.20	2	128	5200	29.94	1.93	2	128	5200	29.94	2.03
		256	4500	34.35	6.68		256	4600	34.35	6.64		256	5400	34.35	6.66
	4	128	2600	27.39	5.37	4	128	3600	27.38	5.21	4	128	4300	27.38	5.43



Fig. 3. Results for the Lena input image of size 128×128 magnified by a factor of $m = 2$. Top row: low-resolution input, bicubic interpolation, Yang *et al.*'s method [16], and Zeyde *et al.*'s method [17]. Bottom row: Chen *et al.*'s method [21], Dong *et al.*'s method [27], our method, and the original HR image.

variations in the PSNR, where ‘‘Clown’’ was the test image of size 128×128 with variable SMSE thresholds. The results show that the total computation time was reduced greatly and the PSNR remained constant as the SMSE threshold increased for each pseudo-HR image patch.

The experimental results obtained for $m = 2$ are presented in Tables I and II, where the results shown in Table I were based on input images of size 128×128 whereas those in Table II had input images of 256×256 pixels. Table III shows the reconstruction results for $m = 4$ with input images of size 128×128 . Each result in Tables I, II, and III represents the average of 10 independent runs. In these three tables, it should be noted that the results obtained using Yang *et al.*'s method and Zeyde *et al.*'s method only include the reconstruction phase time and not the dictionary pair learning process. With the same magnification m and LR patch size N , both methods simply trained the dictionary pair once but the time required for dictionary pair learning was far longer than that using our method. For the elapsed times required using our proposed approach in Tables I, II, and III, the figures outside the parentheses indicate the reconstruction phase time and those

in parentheses are the total computation time, including the dictionary learning and reconstruction phases. Visual comparisons of the HR reconstruction of images ‘‘Lena,’’ ‘‘Barbara,’’ ‘‘Gold Hill,’’ and ‘‘Mandrill’’ of size 256×256 using various algorithms are shown in Figs. 3-6, with a magnification factor of $m = 2$. Visual comparisons of the HR reconstructions of the ‘‘Girl’’ and ‘‘Peppers’’ images of size 512×512 using various algorithms are shown in Figs. 7 and 8 with a magnification factor of $m = 4$. The results demonstrate that Yang *et al.*'s sparse representation SR reconstruction algorithm performed slightly better in terms of the visual effect compared with our proposed method with a magnification factor of $m = 2$, which was because a meaningful external HR images training set was used where the resolution was similar to the reconstructed image, although our algorithm was far more efficient than Yang's in terms of speed. Compared with Chen *et al.*'s MH prediction SR reconstruction algorithm, which only uses the LR input image without an external HR images training set for single image SR reconstruction, the performance of our method was very similar in terms of the visual effect but far superior in terms of efficiency. Although Zeyde *et al.*'s



Fig. 4. Results for the Barbara input image of size 128×128 magnified by a factor of $m = 2$. Top row: low-resolution input, bicubic interpolation, Yang *et al.*'s method [16], and Zeyde *et al.*'s method [17]. Bottom row: Chen *et al.*'s method [21], Dong *et al.*'s method [27], our method, and the original HR image.



Fig. 5. Results for the Gold Hill input image of size 128×128 magnified by a factor of $m = 2$. Top row: low-resolution input, bicubic interpolation, Yang *et al.*'s method [16], and Zeyde *et al.*'s method [17]. Bottom row: Chen *et al.*'s method [21], Dong *et al.*'s method [27], our method, and the original HR image.

method has significantly reduced computational complexity compared with Yang *et al.*'s method, it is still not as efficient as our method in terms of reconstruction time. Dong *et al.*'s method achieved the best reconstruction performance in most cases since it can suppress the sparse coding noise; however, the method is the most computationally expensive one which on average costs more than a hundred times of the reconstruction time of our method. With a magnification factor of $m = 4$, our proposed method performed better in terms of the reconstruction effect and computation time. However, our learned dictionary, which effectively avoids the

complex dictionary training procedure required by Yang *et al.*'s approach [16], provides a more compact representation of patch pairs.

In our experiments, we found that the highest SMSE thresholds differed for the images. However, we can select a suitable threshold for most images to maintain the quality and efficiency of batch image reconstruction. Table IV demonstrates how the size of the input image, the number n of image patches sampled from pseudo-HR images, and the SMSE threshold λ affect the PSNR and total computation time, including dictionary learning and reconstruction, with

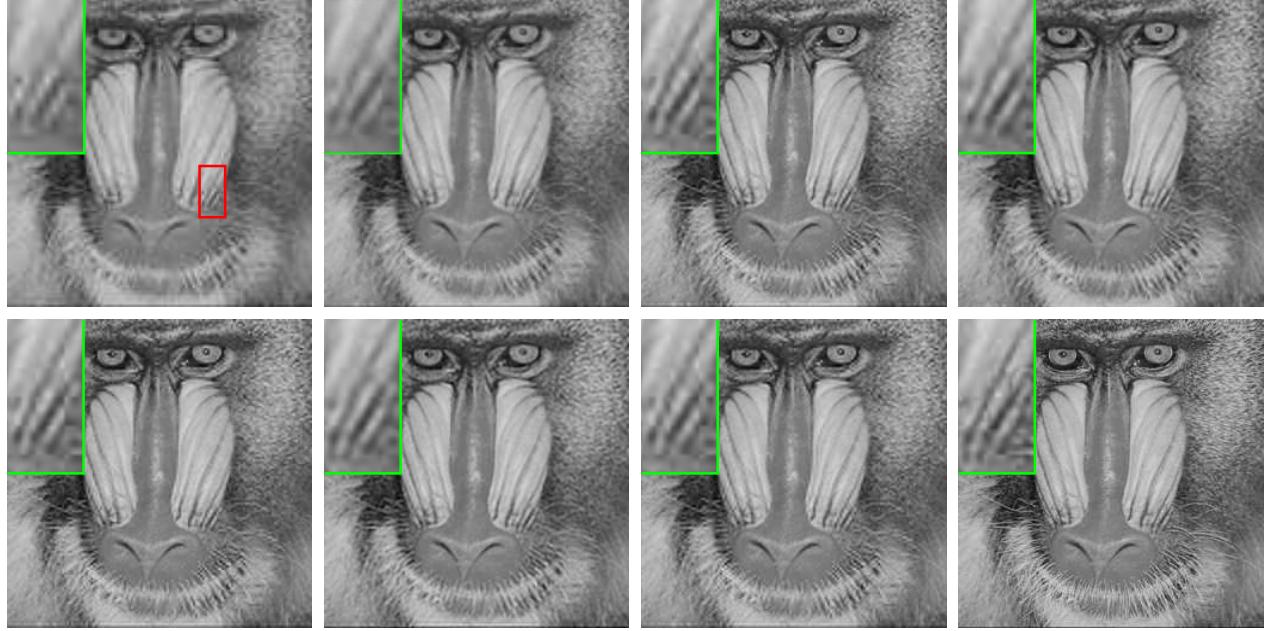


Fig. 6. Results for the Mandrill input image of size 128×128 magnified by a factor of $m = 2$. Top row: low-resolution input, bicubic interpolation, Yang *et al.*'s method [16], and Zeyde *et al.*'s method [17]. Bottom row: Chen *et al.*'s method [21], Dong *et al.*'s method [27], our method, and the original HR image.

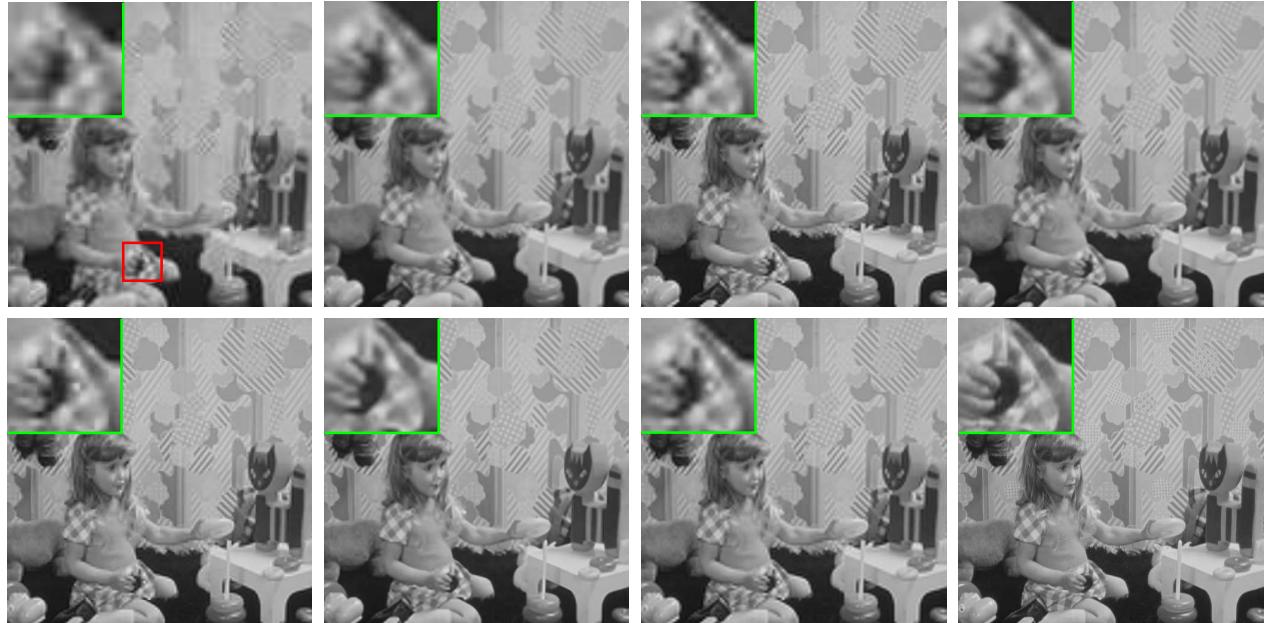


Fig. 7. Results for the Girl input image of size 128×128 magnified by a factor of $m = 4$. Top row: low-resolution input, bicubic interpolation, Yang *et al.*'s method [16], and Zeyde *et al.*'s method [17]. Bottom row: Chen *et al.*'s method [21], Dong *et al.*'s method [27], our method, and the original HR image.

different magnification factors based on three images.

We also evaluated the effect of the dictionary size on single image SR using our proposed approach. We magnified three input images of size 256×256 by a magnification factor of $m = 2$ and three input images of size 128×128 by a magnification factor of $m = 4$, where $N = 2$ in both processes, to determine the optimal size of the HR dictionary D_h , as shown in Tables V and VI. The results show that the speed of dictionary learning and reconstruction would decline when the dimensionality Z of atoms in the dictionary is not $m^2 N^2$, i.e., $Z \neq m^2 N^2$. Therefore, we can conclude that

the dictionary may be most suitable for reconstructing HR images when $Z = q$, regardless of PSNR, RMSE, SSIM, or the total computation time. Thus, all the SR processes conducted using our proposed approach were implemented based on a HR dictionary D_h , where the size was $q \times q$ because of $Z = q$, and we obtained an LR dictionary D_l of size $p \times q$ by bicubic interpolation from D_h .

IV. CONCLUSION

In this paper, we proposed a novel algorithm for fast single image SR based on self-example patch-based dictionary learn-

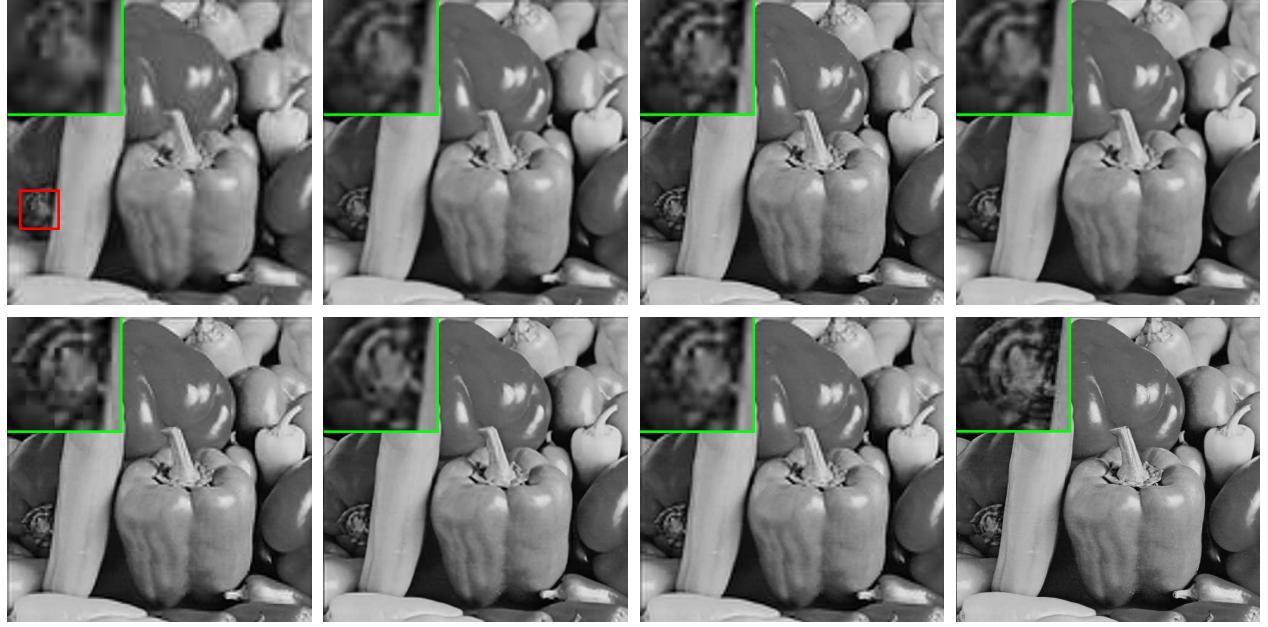


Fig. 8. Results for the Peppers input image of size 128×128 magnified by a factor of $m = 4$. Top row: low-resolution input, bicubic interpolation, Yang *et al.*'s method [16], and Zeyde *et al.*'s method [17]. Bottom row: Chen *et al.*'s method [21], Dong *et al.*'s method [27], our method, and the original HR image.

Algorithm 1 Fast Single Image SR via Self-Example and Sparse Representation

1. **Input:** an LR image Y , magnification factor m , threshold of SMSE λ .
2. **Initialization:** the initial pseudo-HR image $X_m = \text{Bicubic}(Y)$, random normalized dictionary D_h .
3. Sample n patches of size $mN \times mN$ from X_m and obtain the image patch column vectors set X_h under λ .
4. Repeat until convergence (complete learning of D_h):
 - Sparse Coding Stage: Fix D_h and use OMP algorithm to obtain the sparse coefficients β with X_h in (6).
 - Dictionary Update Stage: **For** $j = 1$ to Z ,
 - Update D_h and β using (8).
 - Until** objective function in (7) remains unchanged.
5. **For** each $N \times N$ LR patch of y of Y , starting from the upper-left corner where o pixels overlap in each direction,
 - Use OMP algorithm to solve the sparse coefficients α in (9).
 - Generate the HR image patch \hat{I}_{HR} in (10) and put this patch into an HR image \hat{X} .
6. Using gradient descent, find the closest image to \hat{X} that satisfies the reconstruction constraint defined in (11):

$$X^* = \arg \min_X \|S \cdot X - Y\|_2^2 + c\|X - \hat{X}\|_2^2$$

7. **Output:** HR image X^* .
-

ing and sparse representation. Our proposed strategy exploits the sparse signal representation theory in the framework of CS and dictionary learning of image patches. No HR training set is required for our SR method, in which we exploit image patches within a single image and sparse representation, with only one learned dictionary. This makes our method more practical than

TABLE V
PSNR (DB), RMSE, SSIM, AND TOTAL COMPUTATION TIME (S) RESULTS FOR RECONSTRUCTED IMAGES (512×512) USING DICTIONARIES OF DIFFERENT SIZES Z , SCALE FACTOR $m = 2$, $N = 2$, THRESHOLD OF SMSE $\lambda = 1500$, AND PATCH NUMBER $n = 3000$.

$m = 2, N = 2, \lambda = 1500, n = 3000$					
Image	Z	PSNR	RMSE	SSIM	TIME
Lena	16	35.81	4.15	0.9979	6.67
	32	35.81	4.16	0.9979	8.53
	64	35.80	4.16	0.9979	12.22
	128	35.80	4.16	0.9979	19.77
	256	35.80	4.16	0.9979	35.85
	512	35.80	4.16	0.9979	75.52
	1024	35.80	4.16	0.9979	286.47
	2048	35.80	4.16	0.9979	1126.73
Girl	16	38.05	3.02	0.9983	6.92
	32	38.05	3.02	0.9983	8.61
	64	38.05	3.02	0.9983	12.44
	128	38.05	3.02	0.9983	19.94
	256	38.05	3.02	0.9983	35.36
	512	38.05	3.02	0.9983	74.79
	1024	38.05	3.02	0.9983	285.09
	2048	38.05	3.02	0.9983	1006.72
Peppers	16	32.66	5.94	0.9975	6.82
	32	32.66	5.94	0.9975	8.73
	64	32.66	5.94	0.9975	12.38
	128	32.66	5.94	0.9975	20.72
	256	32.66	5.94	0.9975	36.64
	512	32.66	5.94	0.9975	75.40
	1024	32.66	5.94	0.9975	291.31
	2048	32.66	5.94	0.9975	1105.48

competing SR approaches that use external HR training sets because there is no guarantee that a relevant HR training set is available for LR input images in all situations. Compared with other SR algorithms, our proposed approach is highly competitive in terms of the reconstruction performance but far superior in terms of computational efficiency for natural images.

TABLE VI
PSNR (DB), RMSE, SSIM, AND TOTAL COMPUTATION TIME (S) RESULTS
FOR RECONSTRUCTED IMAGES (512×512) USING DICTIONARIES OF
DIFFERENT SIZES Z , SCALE FACTOR $m = 4$, $N = 2$, THRESHOLD OF
SMSE $\lambda = 1500$, AND PATCH NUMBER $n = 3000$.

$m = 4, N = 2, \lambda = 1500, n = 3000$					
Image	Z	PSNR	RMSE	SSIM	TIME
Lena	16	29.58	8.46	0.9351	14.34
	32	29.58	8.46	0.9351	8.19
	64	29.58	8.46	0.9351	6.77
	128	29.58	8.46	0.9351	8.24
	256	29.58	8.46	0.9351	14.45
	512	29.58	8.46	0.9351	38.62
	1024	29.58	8.46	0.9351	223.13
	2048	29.58	8.46	0.9351	1178.29
Girl	16	31.82	6.53	0.9383	14.03
	32	31.82	6.53	0.9383	7.96
	64	31.82	6.53	0.9383	6.21
	128	31.82	6.53	0.9383	8.21
	256	31.82	6.53	0.9383	14.55
	512	31.82	6.53	0.9383	36.71
	1024	31.82	6.53	0.9383	214.50
	2048	31.82	6.53	0.9383	1098.32
Peppers	16	28.64	9.43	0.9528	15.01
	32	28.64	9.43	0.9528	8.90
	64	28.64	9.43	0.9528	8.13
	128	28.64	9.43	0.9528	8.95
	256	28.64	9.43	0.9528	15.16
	512	28.64	9.43	0.9528	39.77
	1024	28.64	9.43	0.9528	235.44
	2048	28.64	9.43	0.9528	1212.19

REFERENCES

- [1] S. C. Park, M. K. Park and M. G. Kang, "Super-resolution image reconstruction: A technical overview," *IEEE Signal Processing Magazine*, vol. 20, no. 3, pp. 21-36, May. 2003.
- [2] S. Baker and T. Kanade, "Limits on super-resolution and how to break them," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 9, pp. 1167-1183, Sep. 2002.
- [3] M. T. Merino and J. Nunez, "Super-resolution of remotely sensed images with variable-pixel linear reconstruction," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 5, pp. 1446-1457, May. 2007.
- [4] X. Gao, K. Zhang, D. Tao and X. Li, "Joint learning for single-image super-resolution via a coupled constraint," *IEEE Transactions on Image Processing*, vol. 21, no. 2, pp. 469-480, Feb. 2012.
- [5] S. Farsiu, M. D. Robinson, M. Elad and P. Milanfar, "Fast and robust multiframe super-resolution," *IEEE Transactions on Image Processing*, vol. 13, pp. 1327-1344, Oct. 2004.
- [6] M. E. Tipping and C. M. Bishop, "Bayesian image super-resolution," in *Advances in Neural Information and Processing Systems 16 (NIPS)*, 2003.
- [7] X. Gao, K. Zhang, D. Tao and X. Li, "Image super-resolution with sparse neighbor embedding," *IEEE Transactions on Image Processing*, vol. 21, no. 7, pp. 3194-3205, Jul. 2012.
- [8] S. Dai, M. Han, W. Xu, Y. Wu and Y. Gong, "Soft edge smoothness prior for alpha channel super resolution," in *IEEE Conference on Computer Vision and Pattern Classification (CVPR)*, Minneapolis, MN, Jun. 2007, pp. 1-8.
- [9] J. Sun, Z. Xu and H. Shum, "Image super-resolution using gradient profile prior," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Anchorage, AK, Jun. 2008, pp. 1-8.
- [10] S. Farsiu, M. D. Robinson, M. Elad and P. Milanfar, "Fast and robust multiframe super resolution," *IEEE Transactions on Image Processing*, vol. 13, no. 10, pp. 1327-1344, Oct. 2004.
- [11] D. Capel and A. Zisserman, "Super-resolution from multiple views using learnt image models," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Kauai, HI, Dec. 2001, pp. 627-634.
- [12] Q. Yuan, L. Zhang and H. Shen, "Multiframe super-resolution employing a spatially weighted total variation model," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 3, pp. 379-392, Mar. 2012.
- [13] S. Baker and T. Kanade, "Hallucinating faces," in *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, Grenoble, France, Mar. 2000, pp. 83-88.
- [14] W. T. Freeman, T. R. Jones and E. C. Pasztor, "Example-based super-resolution," *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 56-65, Mar./Apr. 2002.
- [15] K. Zhang, X. Gao, D. Tao and X. Li, "Multi-scale dictionary for single image super-resolution," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, RI, Jun. 2012, pp. 1114-1121.
- [16] J. Yang, J. Wright, T. Huang and Y. Ma, "Image super-resolution via sparse representation," *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2861-2873, Nov. 2010.
- [17] R. Zeyde, M. Elad and M. Protter, "On single image scale-up using sparse-representations," in *7th International Conference on Curves and Surfaces*, 2010, pp. 711-730.
- [18] H. Zhang, Y. Zhang and T. Huang, "Efficient sparse representation based image super resolution via dual dictionary learning," in *IEEE International Conference on Multimedia and Expo (ICME)*, Barcelona, Spain, Jul. 2011, pp. 1-6.
- [19] D. Glasner, S. Bagon and M. Irani, "Super-resolution from a single image," in *IEEE International Conference on Computer Vision (ICCV)*, Kyoto, Japan, Sep. 2009, pp. 349-356.
- [20] N. Suetake, M. Sakano and E. Uchino, "Image super-resolution based on local self-similarity," *Optical Review*, vol. 15, no. 1, pp. 26-30, 2008.
- [21] C. Chen and J. E. Fowler, "Single-image super-resolution using multi-hypothesis prediction," in *Proceedings of the 46th Asilomar Conference on Signals, Systems and Computers*, Nov. 2012, pp. 608-612.
- [22] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289-1306, 2006.
- [23] C. Chen, E. W. Tramel and J. E. Fowler, "Compressed-sensing recovery of images and video using multihypothesis predictions," in *Proceedings of the 45th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, Nov. 2011, pp. 1193-1198.
- [24] C. Chen, W. Li, E. W. Tramel and J. E. Fowler, "Reconstruction of hyperspectral imagery from random projections using multihypothesis prediction," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 1, pp. 365-374, Jan. 2014.
- [25] C. Chen, W. Li, E. W. Tramel, M. Cui, S. Prasad, and J. E. Fowler, "Spectral-Spatial Preprocessing Using Multihypothesis Prediction for Noise-Robust Hyperspectral Image Classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 4, pp. 1047-1059, Apr. 2014.
- [26] C. Chen, W. Li, H. Su, and K. Liu, "Spectral-Spatial Classification of Hyperspectral Image based on Kernel Extreme Learning Machine," *Remote Sensing*, vol. 6, no. 6, pp. 5795-5814, June 2014.
- [27] W. Dong, L. Zhang, G. Shi and X. Li, "Nonlocally centralized sparse representation for image restoration," *IEEE Transactions on Image Processing*, vol. 22, no. 4, pp. 1620-1630, Apr. 2013.
- [28] Z. Pan, J. Yu, H. Huang, S. Hu, A. Zhang, H. Ma and W. Sun, "Super-resolution based on compressive sensing and structural self-similarity for remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 9, pp. 4864-4876, Sep. 2013.
- [29] J. F. Murray and K. Kreutz-Delgado, "Learning sparse overcomplete codes for images," *The Journal of VLSI Signal Processing*, vol. 45, pp. 97-110, 2007.
- [30] N. Fan, "Super-resolution using regularized orthogonal matching pursuit based on compressed sensing theory in the wavelet domain," in *IEEE Conference on Computer Graphics, Imaging and Visualization (CGIV)*, Tianjin, Aug. 2009, pp. 349-354.
- [31] M. Aharon, M. Elad and A. Bruckstein, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311-4322, 2006.
- [32] Z. Wang and A. C. Bovik, "Mean squared error: love it or leave it? - a new look at signal fidelity measures," *IEEE Signal Processing Magazine*, vol. 26, no. 1, pp. 98-117, Jan. 2009.
- [33] W. C. Ogle, H. E. Witzgall, M. A. Tinston, J. S. Goldstein and P. A. Zulch, "Independent sample mean squared error for adaptive detection statistics," in *IEEE Aerospace Conference*, Big Sky, MT, Mar. 2005, pp. 2222-2227.
- [34] S. Yang, F. Sun, M. Wang, Z. Liu and L. Jiao, "Novel super resolution restoration of remote sensing images based on compressive sensing and example patches-aided dictionary learning," in *International Workshop on Multi-Platform/Multi-Sensor Remote Sensing and Mapping (M2RSM)*, 2011, pp. 1-6.
- [35] R. Rubinstein, M. Zibulevsky, and M. Elad, "Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit," *Technical Report-CS Technion*, Apr. 2008.
- [36] D. L. Donoho, "For most large underdetermined systems of linear equations, the minimal ℓ^1 -norm solution is also the sparsest solution,"

- Communications on Pure and Applied Mathematics*, vol. 59, no. 6, pp. 797-829, 2006.
- [37] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600-612, April 2004.



Zhiliang Zhu received the M. S. degree in Computer Applications and the Ph. D degree in Computer Science from the Northeastern University, Shenyang, China. He is a Fellow of the China Institute of Communications.

His main research interests include information integrate, complexity software system, network coding and communication security, chaos-based digital communications, applications of complex-network theories, and cryptography. By far, he has authored and co-authored over 130 international journal pa-

pers and 100 conference papers. Additionally, he published 5 books, including *Introduction to Communication and Program Designing of Visual Basic .NET*. He is also the recipient of 9 academic awards at the national, ministerial and provincial level.

Prof. Zhu has served in different capacities in many international journals and conferences. Currently he serves as a Co-Chair for the 1st-7th International Workshop on Chaos-Fractals Theories and Applications. He is a senior member of the Chinese Institute of Electronics and the Teaching Guiding Committee for Software Engineering under the Ministry of Education.



Chen Chen received the B.E. degree in automation from Beijing Forestry University, Beijing, China, in 2009 and the M.S. degree in electrical engineering from Mississippi State University, Starkville, MS, in 2012.

He is currently working toward the Ph.D. degree in the Department of Electrical Engineering, University of Texas at Dallas, Richardson, TX. His research interests include compressed sensing, signal and image processing, pattern recognition, computer vision, and hyperspectral image analysis. He is an active reviewer for the *IEEE Transactions on Image Processing*, *IEEE Transactions on Human-Machine Systems*, the *IEEE Signal Processing Letters*, and the *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*. More information can be found in his personal website <https://sites.google.com/site/chenresearchsite/>.



Fangda Guo received the B.E. degree in Software Engineering from the Northeastern University, Shenyang, China, in 2013. He is currently working toward the double M.S. degrees in the College of Software, Northeastern University, Shenyang, China, and the Department of Computer Engineering and Systems Science, University of Pavia, Lombardy, Italy.

His research interests include compressed sensing, signal and image processing, machine learning, computer vision, pattern recognition, and human-

computer interaction.



Hai Yu received the B.E. degree in Electronic Engineering from the Jilin University, Changchun, China, in 1993 and the Ph. D degree in Computer Software and Theory from the Northeastern University, Shenyang, China, in 2002. He is presently Associate Professor of Software Engineering at the Northeastern University, China.

His research interests include multimedia security and secure chaos-based communications, information hiding, video coding, digital chaotic cipher, network coding, channel coding, and applications of complex-network theories. He has published more than 40 international journal papers and 30 conference papers.

Currently he serves as an Associate Editor for the International Journal and Bifurcation and Chaos, Guest Editor for the Entropy and Guest Editor for the Journal of Applied Analysis and Computation. In addition, he was also a Leader Guest Editor for the Mathematical Problems in Engineering in 2013. As well as, he served in a few international conferences in different roles, such as the Associate Chair for the 7th IWCFTA in 2014, the Program committee Chair for the 4th IWCFTA in 2010, the Chair of Best Paper Award Committee of the 9th International Conference for Young Computer Scientists in 2008, and Program committee member for the 3rd-7th IWCFTA and the 5th Asia-Pacific Workshop on Chaos Control and Synchronization.