

Compressibility Constrained Sparse Representation With Learnt Dictionary for Low Bit-Rate Image Compression

Mai Xu, *Member, IEEE*, Shengxi Li, *Student Member, IEEE*, Jianhua Lu, *Senior Member, IEEE*, and Wenwu Zhu, *Fellow, IEEE*

Abstract—This paper proposes a compressibility constrained sparse representation (CCSR) approach to low bit-rate image compression using a learnt over-complete dictionary of texture patches. Conventional sparse representation approaches for image compression are based on matching pursuit (MP) algorithms. Actually, the weakness of these approaches is that they are not stable in terms of sparsity of the estimated coefficients, thereby resulting in the inferior performance in low bit-rate image compression. In comparison with MP, convex relaxation approaches are more stable for sparse representation. However, it is intractable to directly apply convex relaxation approaches to image compression, as their coefficients are not always compressible. To utilize convex relaxation in image compression, we first propose in this paper a CCSR formulation, imposing the compressibility constraint on the coefficients of sparse representation for each image patch. In addition, we work out the CCSR formulation to obtain sparse and compressible coefficients, through recursively solving the ℓ_1 -norm optimization problem of sparse representation. Given these coefficients, each image patch can be represented by the linear combination of texture elements encoded in an over-complete dictionary, learnt from other training images. Finally, low bit-rate image compression can be achieved, owing to the sparsity and compressibility of coefficients by our CCSR approach. The experimental results demonstrate the effectiveness and superiority of the CCSR approach on compressing the natural and remote sensing images at low bit-rates.

Index Terms—Image compression, over-complete dictionary, sparse representation.

NOMENCLATURE

N	Number of pixels of input image, i.e., image size.
\mathbf{X}	Input image arranged as a row vector, using raster-scan order, $\mathbf{X} \in \mathbb{R}^N$.
K	Number of patches extracted from the input image.

Manuscript received April 15, 2013; revised October 12, 2013, December 3, 2013, January 17, 2014, and February 18, 2014; accepted April 9, 2014. Date of publication April 17, 2014; date of current version October 1, 2014. This work was supported in part by the National Science Foundation of China under Grant 61202139 and in part by the China 973 Program under Grant 2013CB329006. This paper was recommended by Associate Editor P. Salama.

M. Xu and S. Li are with the School of Electronic and Information Engineering, Beihang University, Beijing 100191, China (e-mail: maixu@buaa.edu.cn).

J. Lu is with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China.

W. Zhu is with the Department of Computer Science, Tsinghua University, Beijing 100084, China.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCSVT.2014.2317886

n	Number of pixels ($n = q \times q$ pixels) in each image patch, i.e., patch size.
$\{\mathbf{x}^k\}_{k=1}^K$	Fixed-size image patches arranged as row vectors, extracted from the input image, $\mathbf{x}^k \in \mathbb{R}^n$.
l	Number of elements in an over-complete dictionary.
\mathbf{D}	Learnt over-complete dictionary for sparse representation, $\mathbf{D} \in \mathbb{R}^{n \times l}$.
$\mathbf{m} = \{m^k\}_{k=1}^K$	Mean intensity values of image patches $\{\mathbf{x}^k\}_{k=1}^K$.
$\mathbf{y} = \{\mathbf{y}^k\}_{k=1}^K$	Texture of $\{\mathbf{x}^k\}_{k=1}^K$, obtained by removing mean intensity values $\{m^k\}_{k=1}^K$ from $\{\mathbf{x}^k\}_{k=1}^K$.
$\hat{\mathbf{w}} = \{\hat{\mathbf{w}}^k\}_{k=1}^K$	Coefficients of sparse representation for each image patch, $\hat{\mathbf{w}}^k \in \mathbb{R}^l$.
$\hat{\mathbf{w}}_S$	S nonzero/largest coefficients of $\hat{\mathbf{w}}$.
$\hat{\mathbf{X}} = \{\hat{\mathbf{x}}^k\}_{k=1}^K$	Reconstructed image.

I. INTRODUCTION

IMAGE compression aims to reduce the data size of images and to store or transmit these images efficiently. Therefore, it offers the promise of image/video transmission under limited bandwidth. The past few decades have witnessed an extensive body of literature on both lossless [2] and lossy image compression [3]. In lossy image compression, the central idea is to transform image pixels to obtain compressible coefficients and then only store or transmit the important coefficients, e.g., discrete cosine transform used in Joint Photographic Experts Group (JPEG) [4] and discrete wavelet transform (DWT) used in JPEG 2000 [5]. After obtaining sparse coefficients by quantization, the compression can be achieved via an optimization of encoding nonzero values of the quantized coefficients.

In the early 1990s, research interests of image representation moved from the transform to an over-complete dictionary consisting of a group of mathematical functions called bases or elements. Over-complete means that the number of dictionary elements exceeds the element dimension. At the beginning, matching pursuit (MP) was proposed [6] to decompose a signal by selecting a small subset of functions from a predefined over-complete dictionary. Afterward, Chen and Donoho [7] proposed a basis pursuit approach to decompose a signal

with respect to an over-complete dictionary. As presented in [7], the advantage of basis pursuit is that it may avoid the suboptimality of MP in terms of sparsity. However, the benefit of MP, over basis pursuit, is the high compressibility of its sparse coefficients, which makes it applicable in image/video compression. On the basis of MP, an over-complete dictionary of separable Gabor functions was applied [8] in decomposing motion residual signals for low bit-rate video coding. For image coding, *a posteriori* quantization scheme [9] was proposed for MP coefficients, which decay exponentially when sorted in decreasing order of magnitude. Recently, for low bit-rate image compression, Figueras i Ventura *et al.* [10] developed an over-complete dictionary of 2-D features, built by anisotropic refinement and rotation of contour-like functions. Such a dictionary has shown the effectiveness in compressing the natural images with MP decomposition.

The above image compression approaches represent the natural image, in light of the predefined over-complete dictionary composed of mathematical functions. However, the mathematical functions usually have limited expressiveness [11], when considering the complexity of image patches.¹ The distinct texture of most image patches is comprised of one or more basic texture patterns, to reflect image content. Olshausen and Field [12] have shown the potential of using machine learning approaches to generalize the over-complete dictionary from a set of training image patches. The elements (also called atoms) of such a learnt over-complete dictionary can be seen as the basic texture patterns of generic natural images, and the sparse representation with such a learnt dictionary has been proved to be very similar to simple-cell receptive field in mammalian primary visual cortex. Sparse representation, also called sparse coding, refers to analyzing and explaining a signal structure/pattern, through the representation of the signal by a linear combination of one or more elementary signal patterns from a dictionary.

Benefitting from the success of [12], sparse representation with the learnt dictionary has been successfully implemented in computer vision [13] and image processing [14]. In particular, several relevant approaches have been proposed in [1] and [15]–[17] for low bit-rate image compression. Most of these methods are based on MP algorithms, and thus suffer from the instability in obtaining the sparse coefficients. Here, the instability means the perturbations of the coefficient sparsity for image compression. Convex relaxation (from ℓ_0 -norm to ℓ_1 -norm) approaches [18] are able to yield more stable solutions to sparse representation. Hence, convex relaxation approaches, together with some dictionary learning algorithms, have been utilized in many tasks of computer vision and image processing. However, using the numerical analysis² of decay rate, we argue in this paper (Section III-B) that applying convex relaxation approaches in low bit-rate image compression is more challenging, since the coefficients produced by these approaches are not easily compressible. Therefore, we propose a compressibility constraint sparse

representation (CCSR) approach with a learnt over-complete dictionary for image compression, to benefit from the stability of the existing convex relaxation approaches, while assuring coefficient compressibility. Then, we show that our CCSR approach is capable of compressing the generic natural images at low bit-rates.

More specifically, we first propose a novel CCSR formulation to make the coefficients of sparse representation not only sparse but also compressible. In the CCSR formulation, we embed the compressibility constraint on sparse coefficients for low bit-rate image compression and consider the compatibility between neighboring patches to relieve the block effect of compressed images, which refers to visible artifacts at patch boundaries caused by uncorrelation of neighboring patches. Once sparse and compressible coefficients are achieved through solving the CCSR formulation, an image patch can be represented by a linear combination of elements, selected from the learnt over-complete dictionary.

Then, a recursive ℓ_1 -norm minimization procedure is presented for solving the proposed CCSR formulation to obtain sparse and compressible coefficients, of which near-zero values may be discarded. Afterward, a rational quantization and entropy coding scheme applied to such coefficients results in good rate-distortion performance at low bit-rates.

Finally, we present a gradient descent algorithm to learn the over-complete dictionary of texture patterns from the training images, with two iterative steps. The first step yields the sparse and compressible coefficients for all training image patches, with the over-complete dictionary being fixed. The second step optimizes the over-complete dictionary using the gradient descent algorithm, to fit the training patches and their corresponding coefficients obtained at the first step. Then, these two steps are run iteratively until convergence. Note that since the compressibility constraint has been considered at the first step, the compressibility is imposed when learning the dictionary. The dictionary built by our approach is able to capture the basic texture patterns of image content, thus showing the promise for image compression at low bit-rates.

The rest of this paper is organized as follows. In Section II, we briefly review the related work of sparse representation with learnt over-complete dictionary. In Section III, we propose the CCSR approach for image compression. Section IV presents the dictionary learning method for our CCSR approach. Section V shows experimental results and Section VI concludes this paper.

II. RELATED WORK ON SPARSE REPRESENTATION WITH LEARNT OVER-COMPLETE DICTIONARY

Sparse representation aims at representing a signal by a linear combination of one or more elementary signal patterns from a dictionary. Often, sparse representation allows the dictionary to be over-complete. The earlier work of sparse representation [6], [7] mainly concentrated on seeking the optimal sparse representation of a signal, using a predefined over-complete dictionary. The idea of learning an over-complete dictionary, which yields the sparse representation for a set of training signals, has been extensively studied [14], [19]–[21].

¹Normally, an image can be divided into several nonoverlapping or overlapping patches with fixed size.

²Since image content is diverse, it is infeasible to conduct an analytical analysis on the decay rate of sparse coefficients of image patches.

For example, Kreutz-Delgado *et al.* [19] proposed a Bayesian model, in cooperation with concave/Schur-concave negative log priors, to learn the over-complete dictionary that is appropriate for obtaining the sparse representation of environmental signals. In their approach, environment refers to statistical generating mechanism for signals.

Sparse representation with learnt over-complete dictionary has been successfully applied in the fields of image processing and computer vision [13]. In computer vision area, Wright *et al.* [22] developed a new algorithm to utilize the theory of sparse representation, for automatic recognition of faces from frontal views with varying expression and illumination. Based on the sparse representation of facial images, they argued that the number of extracted features is much more important than the choice of features for recognition. In image denoising, Aharon *et al.* [14] proposed a novel K-SVD algorithm to effectively learn an over-complete dictionary from natural image patches, to remove the image noise with sparse representation. In super resolution, Yang *et al.* [21], [23] proposed an approach to enforce the similarity of sparse representations between the corresponding low-resolution and high-resolution pairs of image patches, by jointly learning two dictionaries that contain low-resolution and high-resolution elements, respectively. Then, the high-resolution image patch can be produced, by applying the sparse representation of the corresponding low-resolution image patch regarding the learnt dictionary with high-resolution elements. In image compression, Bryta and Elad [15] proposed a K-SVD approach, incorporating orthogonal MP algorithm, toward facial image compression at low bit-rates. Afterward, Zepeda *et al.* [16] proposed to use a novel iteration tuned and aligned dictionary for sparse representation in facial image compression. Most recently, using order recursive MP, Skretting and Engan [17] developed a recursive least squares dictionary learning algorithm (RLS-DLA) to compress generic natural images. Although RLS-DLA performs better than other state-of-the-art dictionary learning approaches (such as K-SVD) on generic image compression, it cannot outperform the conventional DWT-based JPEG 2000 approach.

In this paper, to improve the performance of sparse representation with learnt dictionary in image compression, we present a novel CCSR approach with a recursive ℓ_1 -norm optimization solution, instead of the conventional MP-related algorithms, for stably obtaining sparse and compressible coefficients. Indeed, the coefficient compressibility of our CCSR approach enables the discarding of small-valued coefficients, thus making low bit-rate image compression possible.

III. SPARSE REPRESENTATION WITH OVER-COMPLETE DICTIONARY FOR IMAGE COMPRESSION

For image compression, the framework of the proposed CCSR approach is summarized in Fig. 1. The notations used are described in nomenclature section.

As shown in Fig. 1, to compress images using the proposed approach, an off-line step has to be conducted first. At this off-line step, an over-complete dictionary \mathbf{D} of texture patches needs to be learnt from the training data, using the method

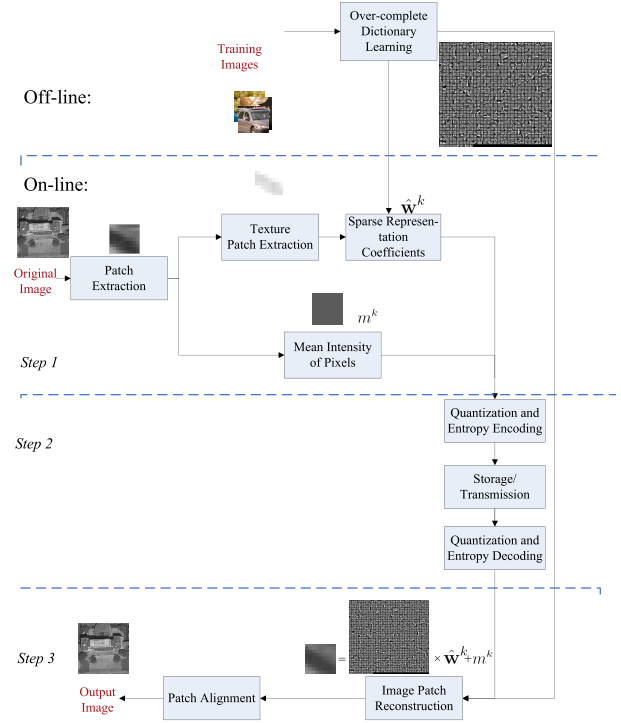


Fig. 1. Framework of the proposed CCSR approach with learnt dictionary for low bit-rate image compression.

introduced in Section IV. Then, the framework of the proposed CCSR approach has the following on-line steps, for compressing target image $\mathbf{X} \in \mathbb{R}^N$ and obtaining reconstructed image $\hat{\mathbf{X}} \in \mathbb{R}^N$.

- 1) *Compression*: This involves calculating mean intensity values \mathbf{m} and seeking sparse and compressible coefficients $\hat{\mathbf{w}}$ of sparse representation with respect to over-complete dictionary \mathbf{D} , for all image patches $\{\mathbf{x}^k\}_{k=1}^K$ with fixed size $(n = q \times q \text{ pixels})$. More details of this step are to be presented in Section III-B.
- 2) *Quantization and Entropy Coding*: This involves conducting quantization and entropy coding of mean intensity values \mathbf{m} and of the nonzero/largest coefficients $\hat{\mathbf{w}}_S$ of $\hat{\mathbf{w}}$. Then, only the encoded bits after entropy coding need to be saved or transmitted to the decoder. For more details, see Section III-C.
- 3) *Decompression*: This involves reconstructing image $\hat{\mathbf{X}}$ via summing mean intensity values \mathbf{m} and the texture patches. Here, the texture patches are built by aggregating the elements of over-complete dictionary \mathbf{D} with their corresponding coefficients in $\hat{\mathbf{w}}$. More details are to be discussed in Section III-D.

Next, the basic idea of sparse representation with over-complete dictionary is introduced, as the foundation of the proposed CCSR approach.

A. Basic Idea of Sparse Representation With Over-Complete Dictionary

Some key problems of signal processing or machine learning can be stated as a linear regression problem, in which

unknown coefficients $\mathbf{w} \in \mathbb{R}^l$ are related to observation $\mathbf{x} \in \mathbb{R}^n$ via

$$\mathbf{x} = \Phi \mathbf{w} + \mathbf{v} \quad (1)$$

where observation matrix $\Phi \in \mathbb{R}^{n \times l}$ is a random measurement matrix in compressive sensing (CS) or an over-complete dictionary in sparse representation, and vector \mathbf{v} normally refers to the noise, caused by the perturbations of the coefficients or the over-complete dictionary.

In most cases, (1) is underdetermined due to the fact that there are more unknown coefficients than the number of equations, i.e., $n < l$. Thus, determining coefficients \mathbf{w} using (1) is an ill-posed problem. As proposed in CS [24], in combination with the prior knowledge that \mathbf{w} is sparse, the sparsest solution of \mathbf{w} to (1) is unique, once observation matrix Φ satisfies the restricted isometry property (RIP). The RIP uses linear algebra to characterize matrices which are closely orthonormal, at least when operating on sparse vectors. Therefore, if the over-complete dictionary for sparse representation satisfies an appropriate RIP or mild RIP condition, then \mathbf{x} can be represented by sparse coefficients \mathbf{w} using (1). From the theoretical point of view, it has been shown in [25] that the over-complete dictionaries also meet the mild condition of RIP. From the empirical point of view, Yang *et al.* [21] has demonstrated that sparse representation with respect to learnt over-complete dictionary of texture, as shown in (1), is capable of representing images for super resolution task. Hence, image compression can be modeled as a linear regression problem to find the sparse coefficients with respect to the learnt over-complete dictionary.

B. Compression: CCSR With Over-Complete Dictionary

In this section, we introduce the compression step for the proposed CCSR approach, in which an image can be decomposed into sparse and compressible coefficients. First of all, patches $\{\mathbf{x}^k\}_{k=1}^K$ are extracted from input image \mathbf{X} in raster-scan order (from left to right and top to bottom). Such an extraction introduces the block effect. To make each image patch compatible with its neighboring patches, we consider the overlap between neighboring patches in both vertical and horizontal directions. Once the number of overlapping pixels increases, the block effect can be reduced along patch boundaries, but at the expense of rate-distortion degradation (because of more patches to be encoded at a given bit-rate). Note that the overlapping size may depend on the patch size, i.e., more overlapping pixels are required for large patches.

Next, to reduce the required size of over-complete dictionary, texture patches $\{\mathbf{y}^k\}_{k=1}^K$ of image patches need to be obtained by removing mean intensity values $\{m^k\}_{k=1}^K$. Then, upon (1), texture patch \mathbf{y}^k can be represented by the following linear regression:

$$\mathbf{y}^k = \mathbf{D} \mathbf{w}^k + \mathbf{v} \quad (2)$$

where $\mathbf{w}^k \in \mathbb{R}^l$ represents the coefficients of sparse representation for the k th image patch and $\mathbf{D} \in \mathbb{R}^{n \times l}$ ($l > n$) is the learnt over-complete dictionary, with each column indicating a texture pattern. For the method of learning \mathbf{D} , see Section IV.

In addition, the ℓ_2 -norm of \mathbf{v} bounds the perturbations of coefficients or dictionary. Then, by computing $\mathbf{w} = \{\mathbf{w}^k\}_{k=1}^K$ and $\mathbf{m} = \{m^k\}_{k=1}^K$, each image patch can be represented via the linear combination of texture elements, selected from over-complete dictionary \mathbf{D} .

As discussed in the above Section III-A, given image patch \mathbf{x}^k and its mean intensity value m^k , determining coefficients \mathbf{w}^k in (2) is an ill-posed problem. However, according to [25], (2) is equal to the following formulation of ℓ_0 -norm minimization:

$$\min_{\mathbf{w}^k} \|\mathbf{w}^k\|_0 \quad \text{s.t.} \quad \|\mathbf{D} \mathbf{w}^k - \mathbf{y}^k\|_2^2 \leq \varepsilon \quad (3)$$

where $\varepsilon = \|\mathbf{v}\|_2^2$ stands for the compression quality and $\min_{\mathbf{w}^k} \|\mathbf{w}^k\|_0$ represents the minimum number of nonzero coefficients. Equation (3) formulates image compression using \mathbf{D} and \mathbf{y}^k . This equation indicates that the motivation of image compression is finding the lowest bit-rate via sparsest coefficients, i.e., $\min_{\mathbf{w}^k} \|\mathbf{w}^k\|_0$, given the compression quality³ $\|\mathbf{D} \mathbf{w}^k - \mathbf{y}^k\|_2^2 \leq \varepsilon$.

Nevertheless, the computational complexity of (3) with the sparsity constraint $\|\mathbf{w}^k\|_0$ is generally overwhelming, as it is an NP-hard problem. To approximately solve this NP-hard problem, several approaches concentrate on seeking for the sparsest solution to (3). One set of approaches are the MP-related approaches, including [6] and [8], and so on, which are simple yet effective. However, as analyzed in [7], MP is not reliable in obtaining the global optimal solution to (3), in terms of sparsity. The other group of approaches are convex relaxation from ℓ_0 -norm to ℓ_1 -norm, e.g., Lasso [18]. These methods provide stability with small perturbations of the coefficient sparsity, at the expense of the computational complexity. In this paper, we mainly focus on the convex relaxation due to its stability. Therefore, according to [7] we can solve (3) by ℓ_1 -norm minimization

$$\min_{\mathbf{w}^k} \|\mathbf{w}^k\|_1 \quad \text{s.t.} \quad \|\mathbf{D} \mathbf{w}^k - \mathbf{y}^k\|_2^2 \leq \varepsilon. \quad (4)$$

Considering the compatibility between neighboring patches, we use the similar way of [21] in enforcing that the currently estimated (k th) patch is nearly consistent with the previous overlapping image patches, to alleviate the block effect along patch boundaries in the reconstructed image. Such compatibility can be modeled by

$$\|\mathbf{E}(\mathbf{D} \mathbf{w}^k + \mathbf{m}^k) - \mathbf{v}^k\|_2^2 \leq \epsilon \quad (5)$$

where matrix \mathbf{E} extracts the upper and left pixels of each image patch, as the overlapping region between the current image patch and its previously processed neighboring patches, \mathbf{v}^k is the vector of intensity values of overlapping region, output by the previously estimated patches, $\mathbf{m}^k = (m^k, \dots, m^k)$ is the vector of mean intensity values for the k th image patch, with the same dimension as \mathbf{v}^k , and ϵ stands for the compatibility level, controlling the possible blocking artifacts between

³It is intractable to directly set an exact value of ε . However, we can increase or decrease ε by modifying the value of parameter μ , to be discussed in (7) afterward.

neighboring patches. Consequently, to adopt the compatibility of neighboring patches, (4) is modified to

$$\min_{\mathbf{w}^k} \|\mathbf{w}^k\|_1 \quad \text{s.t.} \quad \|\mathbf{D}\mathbf{w}^k - \mathbf{y}^k\|_2^2 \leq \varepsilon \\ \|\mathbf{E}(\mathbf{D}\mathbf{w}^k + \mathbf{m}^k) - \mathbf{v}^k\|_2^2 \leq \epsilon. \quad (6)$$

Following the result of [18], the ℓ_1 -norm optimization of (6) can be relaxed to:

$$\min_{\mathbf{w}^k} \|g(\mathbf{w}^k)\|_2^2 + \mu \|\mathbf{w}^k\|_1 \quad (7)$$

where $g(\mathbf{w}^k) = \left(\frac{\mathbf{D}\mathbf{w}^k}{\sqrt{\lambda}\mathbf{E}(\mathbf{D}\mathbf{w}^k + \mathbf{m}^k)} \right) - \left(\frac{\mathbf{y}^k}{\sqrt{\lambda}\mathbf{v}^k} \right)$. Equation (7) indicates the compromise among the error of approximation to \mathbf{y}^k , the compatibility of neighboring patches, and the sparsity of coefficients. Here, λ (≥ 0) stands for the tradeoff between the compatibility of neighboring patches and the reconstruction error, and μ (≥ 0) balances the coefficient sparsity and the reconstruction error. Note that ε and ϵ vanish at the step from (6) to (7), as they are tuned to parameters μ and λ . Here, we define $\sigma(\mathbf{y}^k)$ as the standard deviation of intensities of pixels within texture patch \mathbf{y}^k and thus $\sigma^{-2}(\mathbf{y}^k)$ reflects the texture smoothness of \mathbf{y}^k . To improve the sparsity of coefficients when representing those smooth patches, μ is proportional to the texture smoothness, thereby chosen to be $c^2\sigma^{-2}(\mathbf{y}^k)$, where c is the constant to adjust reconstruction error level. The setting of μ in (7) implies that at a given error level c , more penalty of sparsity is imposed on the patches that have more smooth texture. This is due to the fact that the patches with smooth texture are likely to have less reconstruction error, even without nonzero-valued coefficient.

Ideally, the estimation of \mathbf{w}^k in (7) needs to be conducted simultaneously for all patches, to avoid the consequence of a bad choice of \mathbf{w}^{k-1} on \mathbf{w}^k . Unfortunately, such a kind of optimization cannot be realized due to the complexity constraint. Therefore, \mathbf{w}^k of each patch is estimated in raster-scan order.

Next, p -compressibility is introduced in our approach such that some near-zero coefficients of sparse representation can be discarded for image compression at low bit-rates. p -compressible coefficients mean that when the coefficients are sorted in the order of decreasing magnitude, they decay according to the following power law:

$$|w_{\tau(i)}^k| \leq R \cdot i^{-1/p} \quad (8)$$

where $\tau(i)$ indexes the sorted i th decreasing coefficient of the sparse representation, R is a scaling factor, and p is defined as the decay rate of $|w_{\tau(i)}^k|$. It has been shown in [9] that the compressibility of sparse coefficients output by the MP algorithm is extremely high, as they decay exponentially. Unfortunately, in (7) it is intractable to produce the acceptably compressible coefficients, as shown in Fig. 2. This figure shows that the decay rate of nonzero coefficients produced by (7) is much slower than that of wavelet coefficients, for natural images. Thus, we propose the following formulation to improve the compressibility of sparse coefficients.

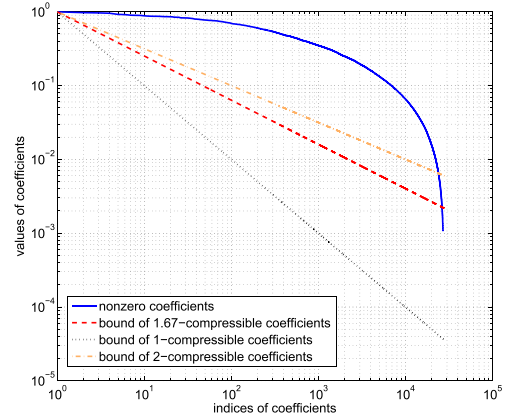


Fig. 2. Average normalized magnitude of sparse coefficients of 10 images randomly selected from the Berkeley image segmentation dataset. Note that the nonzero coefficients of sparse representation were obtained by (7) with convex relaxation. Bounds of various compressibility values are demonstrated as the baselines. It is important to recognize that the wavelet coefficients of natural images are 1.67-compressible as analyzed in [26].

To enforce the nonzero coefficients to be p -compressible, we define a utility function

$$f(w_{\tau(i)}^k) = \begin{cases} |w_{\tau(i)}^k| - R \cdot i^{-1/p}, & \text{if } |w_{\tau(i)}^k| > R \cdot i^{-1/p} \\ 0, & \text{if } |w_{\tau(i)}^k| \leq R \cdot i^{-1/p}. \end{cases} \quad (9)$$

Here, R can be set to $\|\mathbf{w}^k\|_\infty$. In addition, we may have $p = 1.67$, the same as that of wavelet coefficients of natural images, as discussed in [26]. Then, we can jointly optimize the sparsity and compressibility of coefficients by rewriting (7) as

$$\min_{\mathbf{w}^k} \|g(\mathbf{w}^k)\|_2^2 + \left\| \begin{pmatrix} \mu \mathbf{w}^k \\ \gamma f(\mathbf{w}_\tau^k) \end{pmatrix} \right\|_1 \quad (10)$$

where parameter γ (≥ 0) is the importance of coefficient compressibility to the solution and $f(\mathbf{w}_\tau^k)$ is $\{f(w_{\tau(i)}^k)\}_{i=1}^l$, which has been defined in (9). Note that (10) can be reduced to (7) if $f(\mathbf{w}_\tau^k)$ is not considered. Equation (10) combines the compressibility and sparsity of coefficients together in a unified formulation, namely CCSR formulation, thus being able to compress images at various bit-rates, especially at low bit-rates. It is intuitive that for a given bit-rate, less compressibility is required for the coefficients with more sparsity, since the large sparsity, itself, can work out low bit-rate compression. Thus, in this paper we simply set $\gamma = \mu^{-1} = c^{-2}\sigma^2(\mathbf{y}^k)$.

In (10), $\|\gamma f(\mathbf{w}_\tau^k)\|_1$ is added to improve the coefficient compressibility, increasing the limit on the number of coefficients assigned to the complex patches. This may improve the perception of images by encouraging bits allocated to smooth areas with large coefficients, and meanwhile discourages bit wasting on refining complex texture with small coefficients. However, it penalizes the distortion performance, since (10) has a tradeoff between distortion $\|\mathbf{D}\mathbf{w}^k - \mathbf{y}^k\|_2$ and coefficient compressibility $\|f(\mathbf{w}_\tau^k)\|_1$.

Equation (7) can be solved with the coefficients defined as $\tilde{\mathbf{w}}^k$, through a linear programming of ℓ_1 -norm minimization, e.g., by Lasso [18]. Given $\tilde{\mathbf{w}}^k$, the following lemma provides

us with some insights regarding the optimal solution to (10), with a recursive linear programming.

Lemma 1: The ℓ_1 -norm of optimal solution $\hat{\mathbf{w}}^k$ to (10) can be bounded as

$$\|\hat{\mathbf{w}}^k\|_1 \leq \|\tilde{\mathbf{w}}^k\|_1 \leq \frac{\gamma + \mu}{\mu} \|\tilde{\mathbf{w}}^k\|_1 + \frac{1}{\mu} \|g(\tilde{\mathbf{w}}^k)\|_2^2 \quad (11)$$

for a given reconstruction error bound $\|g(\hat{\mathbf{w}}^k)\|_2^2 \leq g(\tilde{\mathbf{w}}^k)\|_2^2 \leq e$.

Proof: Since $\hat{\mathbf{w}}^k$ is the minimal solution to (10), the following result can be obtained:

$$\begin{aligned} \|g(\hat{\mathbf{w}}^k)\|_2^2 + \mu \|\hat{\mathbf{w}}^k\|_1 + \gamma \|f(\hat{\mathbf{w}}^k)\|_1 \\ \leq \|g(\tilde{\mathbf{w}}^k)\|_2^2 + \mu \|\tilde{\mathbf{w}}^k\|_1 + \gamma \|f(\tilde{\mathbf{w}}^k)\|_1. \end{aligned} \quad (12)$$

Given (9), we have $\|f(\tilde{\mathbf{w}}^k)\|_1 \leq \|\tilde{\mathbf{w}}^k\|_1$ and $\|f(\hat{\mathbf{w}}^k)\|_1 \geq 0$. Then, we can obtain

$$\begin{aligned} \|\hat{\mathbf{w}}^k\|_1 &\leq \frac{\gamma + \mu}{\mu} \|\tilde{\mathbf{w}}^k\|_1 + \frac{1}{\mu} (\|g(\tilde{\mathbf{w}}^k)\|_2^2 - \|g(\hat{\mathbf{w}}^k)\|_2^2) \\ &\leq \frac{\gamma + \mu}{\mu} \|\tilde{\mathbf{w}}^k\|_1 + \frac{1}{\mu} \|g(\tilde{\mathbf{w}}^k)\|_2^2. \end{aligned} \quad (13)$$

In addition, since $\tilde{\mathbf{w}}^k$ is defined as the optimal solution to (7), the following inequality holds:

$$\|g(\hat{\mathbf{w}}^k)\|_2^2 + \mu \|\hat{\mathbf{w}}^k\|_1 \geq \|g(\tilde{\mathbf{w}}^k)\|_2^2 + \mu \|\tilde{\mathbf{w}}^k\|_1. \quad (14)$$

If $\|\hat{\mathbf{w}}^k\|_1 < \|\tilde{\mathbf{w}}^k\|_1$, then there exists $\|g(\hat{\mathbf{w}}^k)\|_2^2 > \|g(\tilde{\mathbf{w}}^k)\|_2^2$, according to (14). However, it is inconsistent with error bound $\|g(\hat{\mathbf{w}}^k)\|_2^2 \leq g(\tilde{\mathbf{w}}^k)\|_2^2 \leq e$ of this lemma. Therefore, we can obtain $\|\hat{\mathbf{w}}^k\|_1 \geq \|\tilde{\mathbf{w}}^k\|_1$. \square

Based on Lemma 1, optimal solution $\hat{\mathbf{w}}^k$ to (10), ranging from $\|\tilde{\mathbf{w}}^k\|_1$ to $(\gamma + \mu)/\mu \|\tilde{\mathbf{w}}^k\|_1 + 1/\mu \|g(\tilde{\mathbf{w}}^k)\|_2^2$, can be achieved by recursively solving (7) with conventional linear programming. The detailed procedure is as follows.

- 1) First, the linear programming Lasso is conducted to obtain $\tilde{\mathbf{w}}^k$ as the solution to (7).
- 2) Next, Lasso is run with several trials to solve (7) with a gradually reduced $\tilde{\mu}$ (i.e., reconstruction error bound e decreases gradually) so that the ℓ_1 -norm of the solution $\|\tilde{\mathbf{w}}^k\|_1$ increases from $\|\tilde{\mathbf{w}}^k\|_1$ to $(\gamma + \mu)/\mu \|\tilde{\mathbf{w}}^k\|_1 + 1/\mu \|g(\tilde{\mathbf{w}}^k)\|_2^2$. It is due to the fact that when we reduce $\tilde{\mu}$ for less reconstruction error in (7), the value of $\|\tilde{\mathbf{w}}^k\|_1$ rises. Then, all solutions $\tilde{\mathbf{w}}^k$ of each trial need to be stored. Note that for each trial, we replace μ in (7) by $\tilde{\mu}$, which is initialized to be μ and then is gradually reduced with step $\Delta(\mu) = h \cdot \mu$ (h is a constant, tuned to 0.05 as appropriate).
- 3) Finally, among all stored values of $\tilde{\mathbf{w}}^k$, the one minimizing (10) needs to be output as approximately optimal solution $\hat{\mathbf{w}}^k$. This is due to the fact that the range of $\tilde{\mathbf{w}}^k$ is the same as that of $\hat{\mathbf{w}}^k$, since it has been proved in Lemma 1 that $\|\tilde{\mathbf{w}}^k\|_1 \leq \|\hat{\mathbf{w}}^k\|_1 \leq (\gamma + \mu)/\mu \|\tilde{\mathbf{w}}^k\|_1 + 1/\mu \|g(\tilde{\mathbf{w}}^k)\|_2^2$.

Finally, given an approximately optimal solution $\hat{\mathbf{w}}^k$ to (10), image compression can be achieved via discarding small (including zero-valued and near-zero-valued) coefficients.⁴

⁴The threshold on discarding small-valued coefficients is determined by the target bit-rate.

TABLE I
COMPRESSION ALGORITHM OF THE CCSR
APPROACH FOR IMAGE COMPRESSION

– Input: target image \mathbf{X} and over-complete dictionary \mathbf{D} .
– Output: S largest coefficients $\hat{\mathbf{w}}_S$ and mean intensity values \mathbf{m} .
• Extract image patches $\{\mathbf{x}^k\}_{k=1}^K$ in raster-scan order with overlap in both horizontal and vertical directions.
• For each image patch \mathbf{x}^k
1 Compute the mean intensity value of the patch: $m^k = \sum_i x_i^k$, where x_i^k is the intensity of the i th pixel in the patch.
2 Estimate coefficients $\tilde{\mathbf{w}}^k$ of the patch by recursively solving the optimization problem of (10).
End
• Save the indices and values of S largest coefficients $\hat{\mathbf{w}}_S$ from $\{\tilde{\mathbf{w}}^k\}_{k=1}^K$.
• Return $\hat{\mathbf{w}}_S$ and $\mathbf{m} = \{m^k\}_{k=1}^K$.

Then, only the indices and values of S largest coefficients⁵ $\hat{\mathbf{w}}_S$ in $\hat{\mathbf{w}}^k$ and mean intensity values $\mathbf{m} = \{m^k\}_{k=1}^K$ need to be encoded for each image patch, with the quantization and entropy coding scheme discussed in Section III-C. An image can be reconstructed by the coefficients composed of two parts: 1) sparse coefficients and 2) mean intensity values. The compression procedure of the proposed CCSR method is summarized in Table I.

C. Quantization and Entropy Coding for the Proposed Approach

Similar to encoding the DC coefficients in JPEG, we can quantize mean intensity values \mathbf{m} of image patches via dividing by a quantization constant denoted by b , i.e., $\text{round}(\mathbf{m}/b)$. Then, differential pulse-code modulation (DPCM) and Huffman coding need to be applied in encoding the quantized mean intensity values. The dynamic range of the mean intensity values is $[0, 255]$, much smaller than that of the DC coefficients [4] in JPEG. Therefore, the proposed CCSR approach can save some bits when encoding the DPCM-valued mean intensities, in comparison with encoding DPCM values of the DC coefficients in JPEG.

For quantizing the values of S largest sparse coefficients $\hat{\mathbf{w}}_S$, the codebook with d 1-D codewords needs to be learnt first by K-means method [27]. To be more specific, the learning procedure is presented in the following.

- 1) *Initializing the Codebook With d Codewords at Random:* The initial codewords can be randomly selected from the 1-D coefficients of $\hat{\mathbf{w}}_S$.
- 2) *Clustering $\hat{\mathbf{w}}_S$ Around Each Codeword:* All 1-D coefficients in $\hat{\mathbf{w}}_S$ are divided into d clusters, with the least distance to the codeword (i.e., the centroid point) of each cluster.
- 3) *Updating the New Set of Codewords:* The codeword of each cluster is computed by averaging the coefficients belonging to each cluster.
- 4) *Repeating Steps 2 and 3 Until Convergence:* After several iterations over clustering coefficients (Step 2) and updating codewords (Step 3), the final codewords are output, as the codebook for quantizing $\hat{\mathbf{w}}_S$.

⁵Toward the lower bit-rate, some near-zero coefficients can be discarded, since the coefficients are p -compressible.

Given the learnt codebook, each coefficient of $\hat{\mathbf{w}}_S$ is quantized by choosing the nearest matching codeword. Then, only the index of the matching codeword in the codebook is needed for entropy encoding. Here, Huffman coding is applied as the entropy coding scheme to encode the codeword indices, corresponding to the quantized values of $\hat{\mathbf{w}}_S$. Such an entropy coding scheme may save a lot of bits, due to the fact that the quantized values of $\hat{\mathbf{w}}_S$ are distributed nonuniformly, after considering the coefficient compressibility in (10). For an image, a Huffman table has to be generated according to the actual distributions of quantized $\hat{\mathbf{w}}_S$, and then it is embedded as the prefix in the encoded bitstream. Owing to the small size of codebook for quantizing $\hat{\mathbf{w}}_S$, there is little overhead of the bit-rates on encoding the Huffman table. Once a decoder receives the index of a codeword after Huffman decoding, it replaces the index with the associated codeword, for obtaining the dequantized coefficient values. Note that the mapping from codeword indices to their corresponding values in the codebook is included in the prefix of the resulting encoded bitstream. The cost of such a mapping is little in comparison with the entropy coding of coefficients, since there are only a few codewords in the codebook (to be discussed in Section V-B), especially at low bit-rates. At last, the dequantized values of $\hat{\mathbf{w}}_S$ can be obtained at the decoder.

In addition, the number of nonzero coefficients, from patch to patch, needs to be known at the decoder. Similar to JPEG, an end of patch (EOP) marker is applied in the proposed approach to indicate that there does not exist any nonzero coefficient at the current patch. Since our approach targets low bit-rate image compression, three bits, instead of four bits in JPEG, are utilized for encoding the information of EOP. This results in an overhead of 0.037 bpp for 9×9 patches and an overhead of 0.012 bpp for 16×16 patches. In addition, the indices of nonzero coefficients $\hat{\mathbf{w}}_S$ relative to dictionary \mathbf{D} have to be known at the decoder so that the image patches can be reconstructed by aggregating the elements of \mathbf{D} with their corresponding sparse coefficients. Hence, a fixed length coding may be applied to encode the indices of $\hat{\mathbf{w}}_S$, with the bit depth being $\log_2 l$, where l is the element number of \mathbf{D} .

For the output bitstream, the bits of each patch are arranged in descending order of their corresponding coefficients. The tradeoff between bit-rate and reconstruction quality can be controlled by discarding the bits of small (near-zero) coefficients. The remaining bits, belonging to the output bitstream of several largest coefficients, determine the rate-distortion of image compression. Note that only several patches need to be encoded with Huffman and fixed length coding scheme, since only the DPCM-valued mean intensities are encoded, in particular at low bit-rates, for a great number of smooth patches.

D. Decompression: Image Reconstruction With Nonzero Coefficients

In this section, we move to the decompression step, in which the target image can be reconstructed by the sparse coefficients of image patches. First, at the decoder, coefficients of sparse representation $\hat{\mathbf{w}}^k$ and mean intensity value m^k of each image

patch can be extracted from the bitstream of $\hat{\mathbf{w}}_S$ and \mathbf{m} . Then, given $\hat{\mathbf{w}}^k$ and m^k , image patch $\hat{\mathbf{x}}^k$ can be reconstructed using (2) as

$$\hat{\mathbf{x}}^k = \mathbf{D}\hat{\mathbf{w}}^k + m^k. \quad (15)$$

\mathbf{D} is a learnt over-complete dictionary, the same as that of the encoder. Finally, we can obtain image $\hat{\mathbf{X}}$ at the decoder, by combining all reconstructed image patches $\{\hat{\mathbf{x}}^k\}_{k=1}^K$ together in raster-scan order. Note that the ambiguity of the overlapping pixels between neighboring patches can be avoided by averaging their intensity values.

IV. OVER-COMPLETE DICTIONARY LEARNING FOR IMAGE COMPRESSION

Now, the only task left for the proposed CCSR approach is to learn the over-complete dictionary of texture off-line from training images. In this paper, we use a gradient descent algorithm to learn the over-complete dictionary of texture patterns, $\mathbf{D} \in \mathbb{R}^{n \times l}$. Assume that $\mathbf{X}^* = \{\mathbf{x}_1^*, \dots, \mathbf{x}_T^*, \dots, \mathbf{x}_T^*\}$ is the set of training image patches, randomly extracted from training images. Then, training texture patches $\mathbf{Y}^* = \{\mathbf{y}_1^*, \dots, \mathbf{y}_T^*, \dots, \mathbf{y}_T^*\}$ can be obtained by subtracting the mean intensity values from each training image patch. According to (10), the problem of learning \mathbf{D} can be formulated as

$$\begin{aligned} \min_{\mathbf{D}, \mathbf{W}^*} \sum_t \|\mathbf{y}_t^* - \mathbf{D}\mathbf{w}_t^*\|_2^2 + \left\| \left(\gamma_t^* \mu_t^* \mathbf{w}_t^* \right) \right\|_1 \\ \text{s.t. } \|\mathbf{d}_1\|_2 = 1, \dots, \|\mathbf{d}_l\|_2 = 1 \end{aligned} \quad (16)$$

where $\mathbf{W}^* = \{\mathbf{w}_1^*, \dots, \mathbf{w}_T^*, \dots, \mathbf{w}_T^*\}$ is the set of sparse coefficients, corresponding to texture patches \mathbf{Y}^* , with respect to over-complete dictionary $\mathbf{D} = [\mathbf{d}_1, \dots, \mathbf{d}_l]$. Here, $\mathbf{d}_1, \dots, \mathbf{d}_l$ are the columns of \mathbf{D} , seen as the basic texture elements in the over-complete dictionary, and they are all normalized to 1, to eliminate the scaling ambiguity of coefficients. In addition, the settings of μ_t^* and γ_t^* are the same as those of the CCSR formulation in Section III, i.e., $\mu_t^* = c^2 \sigma^{-2}(\mathbf{y}_t^*)$ and $\gamma_t^* = \mu_t^{*-1}$. Note that different from (10), this formulation does not consider any compatibility between neighboring patches, since the (randomly selected) training image patches are independent of each other.

Learning dictionary \mathbf{D} can be modeled as the optimization problem in (16), solved by minimizing the square reconstruction error and enforcing the coefficients sparse and compressible. Equation (16) is convex [21] in terms of \mathbf{D} , with \mathbf{W}^* being fixed. Given a fixed \mathbf{D} , \mathbf{W}^* in (16) can be solved using the method of Section III-B. As such, \mathbf{D} can be learnt with the following two recursive steps.

At the first step, we can estimate coefficients \mathbf{W}^* by

$$\mathbf{W}^* = \arg \min_{\mathbf{W}^*} \sum_t \|\mathbf{y}_t^* - \mathbf{D}\mathbf{w}_t^*\|_2^2 + \left\| \left(\gamma_t^* \mu_t^* \mathbf{w}_t^* \right) \right\|_1 \quad (17)$$

with \mathbf{D} being fixed. \mathbf{D} may be initialized to be a group of randomly selected training texture patches, with each column unit being normalized. Equation (17) can be solved by the recursive linear programming method introduced in Section III-B.

TABLE II
OVER-COMPLETE DICTIONARY LEARNING ALGORITHM FOR THE
PROPOSED IMAGE COMPRESSION APPROACH

– Input: patches of training images $\mathbf{X}^* = \{\mathbf{x}_1^*, \dots, \mathbf{x}_t^*, \dots, \mathbf{x}_T^*\}$.
– Output: over-complete dictionary \mathbf{D} .
• Generate training texture patches $\mathbf{Y}^* = \{\mathbf{y}_1^*, \dots, \mathbf{y}_t^*, \dots, \mathbf{y}_T^*\}$ by subtracting the mean intensity values from each training image patch, $\mathbf{X}^* = \{\mathbf{x}_1^*, \dots, \mathbf{x}_t^*, \dots, \mathbf{x}_T^*\}$.
• Initialize \mathbf{D} with a group of randomly selected training texture patches.
• Normalize ℓ_2 -norm of each column of \mathbf{D} to 1.
• While halting criterion <i>false</i> (e.g. not reaching a fixed number of iterations)
– Update coefficients \mathbf{W}^* by solving the following optimization problem:
$\mathbf{W}^* = \underset{\{\mathbf{w}_1^*, \dots, \mathbf{w}_t^*, \dots, \mathbf{w}_T^*\}}{\operatorname{argmin}} \sum_t \ \mathbf{y}_t^* - \mathbf{D}\mathbf{w}_t^*\ _2^2 + \left\ \begin{pmatrix} \mu_t^* \mathbf{w}_t^* \\ \gamma_t^* f(\mathbf{w}_t^*) \end{pmatrix} \right\ _1$
– While halting criterion <i>false</i> (e.g. not reaching a fixed number of iterations)
1 Update \mathbf{D} via gradient descent:
$\mathbf{D}_j = \mathbf{D}_{j-1} - 2\eta \sum_t (\mathbf{D}_{j-1} \mathbf{w}_t^* \mathbf{w}_t^{*T} - \mathbf{y}_t^* \mathbf{w}_t^{*T})$
2 Renormalize ℓ_2 -norm of each column of \mathbf{D} to 1.
End
• Return \mathbf{D}

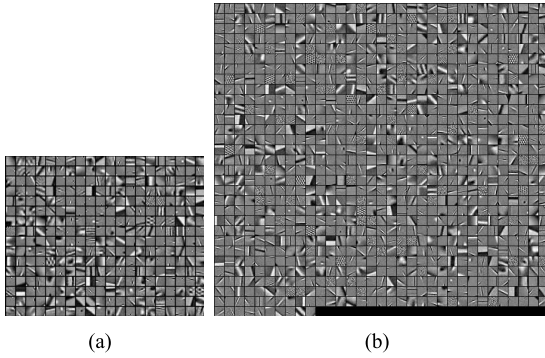


Fig. 3. (a) Learnt over-complete dictionary with element size being 9×9 . (b) Learnt over-complete dictionary with element size being 16×16 . They were both trained from 100000 randomly selected image patches, using the method summarized in Table II. Each small grid with (a) 9×9 pixels and (b) 16×16 pixels, stands for one element of the dictionaries, reflecting a basic texture pattern.

At the second step, we can fix \mathbf{W}^* and then update dictionary \mathbf{D} via

$$\mathbf{D} = \underset{\mathbf{D}}{\operatorname{argmin}} \sum_t \|\mathbf{y}_t^* - \mathbf{D}\mathbf{w}_t^*\|_2^2 \quad \text{s.t. } \|\mathbf{d}_1\|_2 = 1, \dots, \|\mathbf{d}_l\|_2 = 1. \quad (18)$$

Given the above equation, \mathbf{D} can be updated iteratively with gradient descent

$$\mathbf{D}_j = \mathbf{D}_{j-1} - \eta \frac{\partial}{\partial \mathbf{D}} \sum_t \|\mathbf{y}_t^* - \mathbf{D}\mathbf{w}_t^*\|_2^2 \big|_{\mathbf{D}_{j-1}} \quad (19)$$

where \mathbf{D}_j is the value of dictionary \mathbf{D} at the j th iteration, and η ($= 0.01$) is the learning rate of gradient descent. It is proved in the Appendix that

$$\frac{\partial}{\partial \mathbf{D}} \sum_t \|\mathbf{y}_t^* - \mathbf{D}\mathbf{w}_t^*\|_2^2 = 2 \sum_t (\mathbf{D}\mathbf{w}_t^* \mathbf{w}_t^{*T} - \mathbf{y}_t^* \mathbf{w}_t^{*T}). \quad (20)$$

Consequently, (19) can be rewritten as

$$\mathbf{D}_j = \mathbf{D}_{j-1} - 2\eta \sum_t (\mathbf{D}_{j-1} \mathbf{w}_t^* \mathbf{w}_t^{*T} - \mathbf{y}_t^* \mathbf{w}_t^{*T}) \quad (21)$$

TABLE III
PARAMETER SETTINGS OF THE PROPOSED CCSR APPROACH

Step	Parameter	Value
Compression	patch size	9×9 or 16×16
	λ	1
	p	1.67
	c	4
Learning	b	4
	η	0.01
	iteration number of inner loop	100
	iteration number of outer loop	50

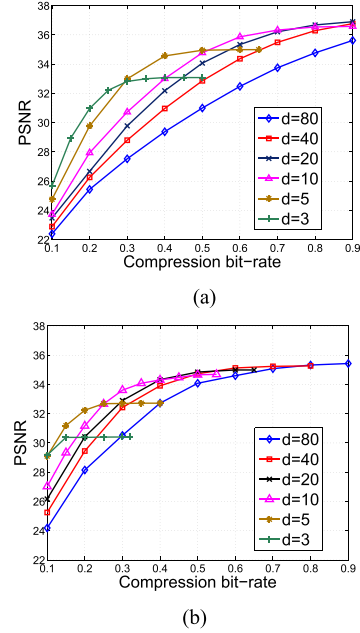


Fig. 4. Curves of the average rate-distortion performance on compressing the standard images of peppers, finishing boat, truck, and couple, using the proposed CCSR approach with varying codebook size d . For a given d , the rate-distortion curve was generated by preserving different amounts of largest coefficients (denoted by S). (a) SR9. (b) SR16. Note that SR16 and SR9 are abbreviations of the proposed CCSR approach with patch sizes being 16×16 and 9×9 , respectively.

where \mathbf{D}_j needs to be iteratively updated toward convergence. In addition, \mathbf{D}_j has to be renormalized at each iteration to make the ℓ_2 -norm of each column of \mathbf{D}_j one. In (21), $\sum_t (\mathbf{D}_{j-1} \mathbf{w}_t^* \mathbf{w}_t^{*T} - \mathbf{y}_t^* \mathbf{w}_t^{*T})$ can be seen as the residual of each iteration.

Then, the above two steps, as summarized in (17) and (21), need to be run recursively until reaching a halting criterion. Table II shows the procedure of learning over-complete dictionary. Note that both the inner and outer loops of Table II are repeated until the halting criteria are triggered. In the experiments of this paper, we merely considered fixed iteration numbers as the halting criteria. Some other stopping rules may also be effective. For example, the outer loop may be terminated once $\sum_t \|\mathbf{y}_t^* - \mathbf{D}\mathbf{w}_t^*\|_2^2 + \mu_t^* \|\mathbf{w}_t^*\|_1 + \gamma_t^* \|f(\mathbf{w}_t^*)\|_1$ is less than a threshold.

V. EXPERIMENTAL RESULTS

In this section, experiments of image compression were performed on extensive images, to validate the proposed CCSR

TABLE IV
SETTINGS OF PARAMETER d IN THE CCSR APPROACH

Patch size	SR9				SR16				
Bit-rate	(0.7, 0.9]	(0.5, 0.7]	[0.3, 0.5]	[0.1, 0.3]	(0.75, 0.9]	(0.55, 0.75]	(0.4, 0.55]	(0.25, 0.4]	[0.1, 0.25]
d	20	10	5	3	80	40	20	10	5

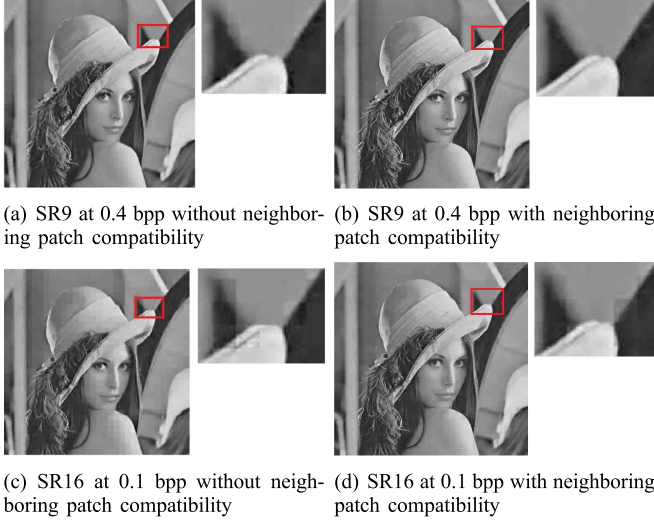


Fig. 5. Reconstructed *Lena* images (resolution: 512×512) of our CCSR approach with and without considering the neighboring patch compatibility. PSNRs of the reconstructed images: (a) 37.66 dB, (b) 37.56 dB, (c) 31.79 dB, and (d) 31.61 dB.

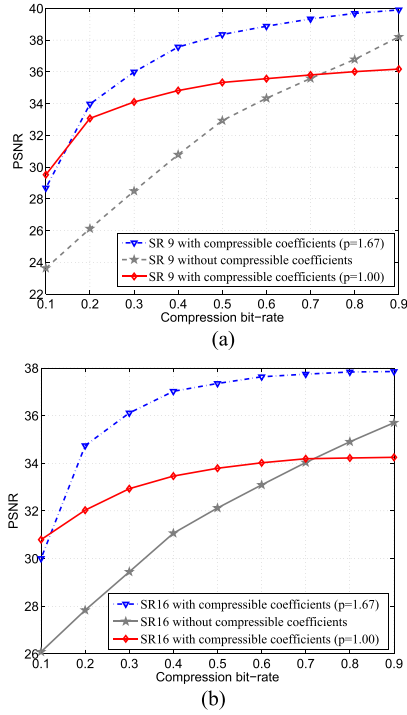


Fig. 6. Rate-distortion comparison of compressing the *Lena* image (resolution: 512×512) with different constraints on coefficient compressibility. (a) SR9. (b) SR16.

approach. For simplicity, we focused on gray images. For the sake of comparison, JPEG 2000, RLS-DLA [17], and MP [10] approaches were employed to compress exactly

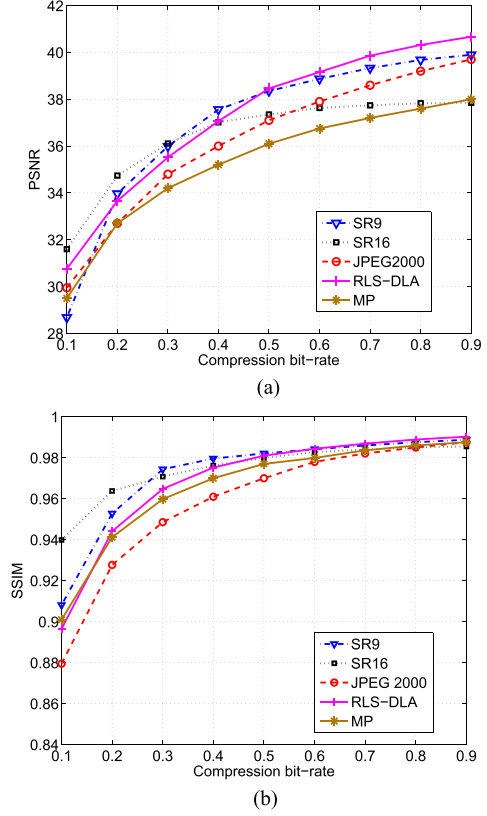


Fig. 7. Rate-distortion curves of compressing the *Lena* images (resolution: 512×512) using JPEG 2000, RLS-DLA, MP, and the proposed CCSR approaches. Note that SR16 and SR9 stand for the proposed CCSR approach with patch sizes being 16×16 and 9×9 , respectively. (a) PSNR. (b) SSIM.

the same images. Here, MP can be seen as a conventional image compression approach, based on sparse representation with predefined dictionary, while RLS-DLA is a state-of-the-art approach, in the spirit of sparse representation with learnt dictionary. Note that the dictionary in the MP approach is independent of training images, while the dictionary in RLS-DLA approach relies on training images. The training images of both RLS-DLA and our CCSR approaches were selected from the Berkeley image database, and they were not used as the testing images. In addition, the parameter settings for RLS-DLA and MP approaches were the same as the ones used in [10] and [17]. For JPEG 2000, the Jasper software [28] was applied, with its default settings.

In the following, Section V-A focuses on learning the prior over-complete dictionary for image compression. In Section V-B, we present the parameter selection of our approach. Then, using the learnt dictionary of Section V-A and parameter settings of Section V-B, Section V-C evaluates the benefits of neighboring patch compatibility and coefficient compressibility, which are introduced by our approach. Finally, Section V-D compares the experimental results of



Fig. 8. *Lena* images (resolution: 512×512) compressed by (a) RLS-DLA, (b) JPEG 2000, (c) MP, and (d) proposed CCSR approaches, at 0.1 bpp.

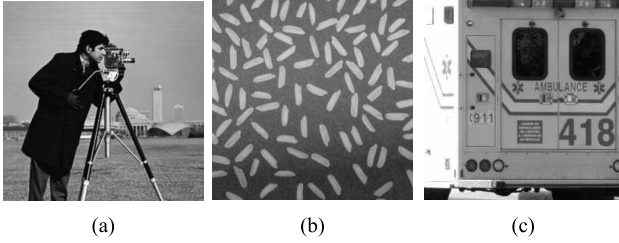


Fig. 9. Three original input images for image compression. The resolutions of all these images are 256×256 . (a) *Camera man*. (b) *Rice*. (c) *Ambulance*.

different approaches on compressing the natural images as well as the remote sensing images.

A. Over-Complete Dictionary Learnt for Image Compression

To compress images with the proposed method, the over-complete dictionary has to be learnt and stored in advance. Here, the over-complete dictionary was learnt from 100 000 training image patches, using the method of Section IV. All these image patches were randomly selected⁶ from 100 images of the Berkeley image database, in which no remote sensing image was involved. Note that in this section, the parameter settings for dictionary learning were the same as those for sparse representation on compressing images, as reported in Section V-B.

Fig. 3 shows the learnt dictionaries. It has been revealed empirically that the optimal selection of the number of elements for a dictionary is ~ 4 times of the patch size (i.e., the dictionary redundancy is 4) for reconstructing facial images [15] and generic natural images [29]. Thereby, the dictionary in Fig. 3(a) contains 320 elements of 9×9 texture patches and the dictionary in Fig. 3(b) includes 1000 elements of 16×16 texture patches. It can be seen from this figure that the basic patterns of texture patches, e.g., the regular surfaces and orientated edges, are encoded in the learnt dictionaries. Since most natural scenes are normally comprised of the objects with various regular texture patterns, the learnt dictionaries are able to well represent natural images.

B. Selection of Parameters

In our experiments, the parameter settings of our CCSR approach are tabulated in Table III. To be more specific,

⁶As the patches were extracted for learning the dictionary of texture patterns, the smooth patches with little variance of pixel intensities were discarded when learning dictionary.

in the dictionary learning step, we chose 100 and 50 as the inner and outer iteration numbers, respectively. In addition, we followed [19] in the setting of $\eta = 0.01$. In the compression step, we applied the same choice of $\lambda = 1$ as the previous work of super resolution [21]. Then, we utilized the decay rate of wavelet coefficients [26] to set $p = 1.67$. We further compared the rate-distortion performance under different values of p in Section V-C (Fig. 6), to show the effectiveness of $p = 1.67$. After obtaining sparse coefficients, the bits produced by quantized mean intensity values after DPCM only take up a small amount of total bits in image compression, but greatly influence the distortion. Therefore, a small value of 4 was chosen for quantization constant b . At last, c was empirically set to 4, for generally yielding the satisfactory results, in terms of rate-distortion performance.

In addition, it is interesting to analyze the tradeoff between the number of coefficients used to represent an image and the average number of bits allocated to each coefficient. In the proposed CCSR approach, the average number of bits allocated to each coefficient relies on the codebook size (denoted by d) during the quantization step. In addition, the number of coefficients used to represent an image in the CCSR approach is controlled by two factors: 1) μ denoted by $c^2 \sigma^{-2}(\mathbf{y}^k)$ for the coefficient sparsity and 2) S largest coefficients preserved for image reconstruction.

Four generic standard images from the SIPI database [30] (including peppers, fishing boat, truck, and couple) were utilized to analyze the tradeoff between the number of coefficients used to represent an image and the average number of bits allocated to each coefficient. Fig. 4 shows the average rate-distortion curves of the proposed CCSR approach on compressing the four standard images for varying codebook size d . This figure implies that d has to be increased along with the rising bit-rate toward the optimal rate-distortion performance. In light of Fig. 4, the settings of d are listed in Table IV, for achieving the empirically optimal rate-distortion performance in SR9 (the proposed CCSR approach with 9×9 patch) and SR16 (the proposed CCSR approach with 16×16 patch).

C. Evaluation of Neighboring Patch Compatibility and Coefficient Compressibility

Since the proposed CCSR approach introduces new terms in (10) to guarantee the neighboring patch compatibility and coefficient compressibility, it is important to evaluate their benefits to low bit-rate image compression. In this section,



Fig. 10. Images compressed by (a) RLS-DLA, (b) JPEG 2000, (c) MP, and (d) proposed CCSR approaches, at 0.4 bpp.

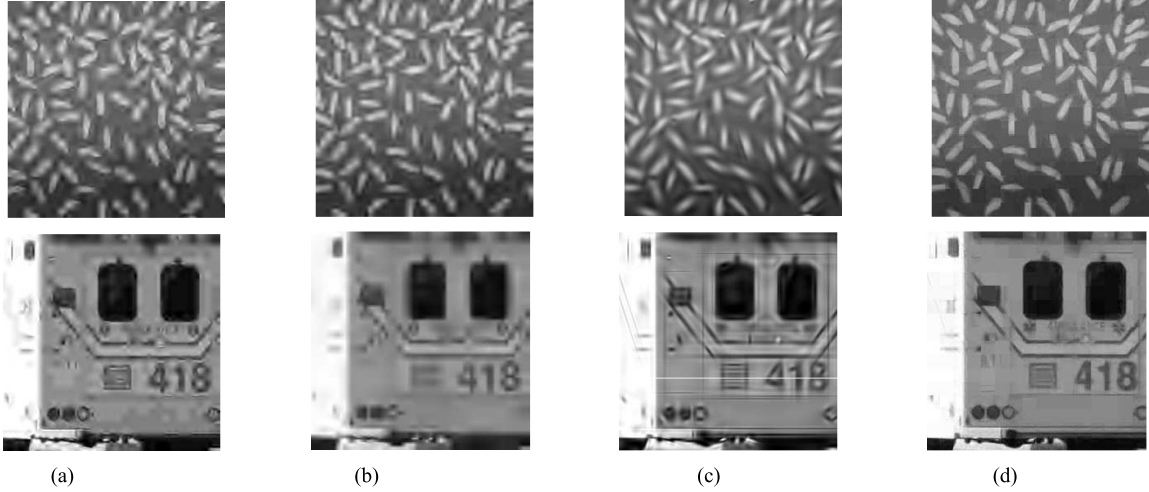


Fig. 11. Images compressed by (a) RLS-DLA, (b) JPEG 2000, (c) MP, and (d) proposed CCSR approaches, at 0.1 bpp.

TABLE V
PSNRs(dB)/SSIMs OF RECONSTRUCTED NATURAL IMAGES BY DIFFERENT COMPRESSION APPROACHES

Images	Resolution	Bit-rate = 0.4 bpp				Bit-rate = 0.1 bpp			
		MP	RLS-DLA	JPEG 2000	CCSR	MP	RLS-DLA	JPEG 2000	CCSR
Camera Man	256 × 256	29.0/0.83	29.8/0.85	30.0/0.86	30.4/0.91	23.9/0.70	24.1/0.70	23.8/0.70	25.1/0.80
Rice	256 × 256	28.4/0.76	28.8/0.77	29.1/0.78	30.3/0.87	24.1/0.63	23.8/0.62	24.0/0.63	25.3/0.70
Ambulance	256 × 256	27.8/0.73	29.3/0.86	28.5/0.77	28.8/0.84	24.6/0.57	24.4/0.66	24.5/0.57	25.7/0.70
Barbara	512 × 512	29.4/0.93	30.3/0.94	30.9/0.94	31.8/0.96	24.4/0.79	24.5/0.79	24.7/0.80	25.1/0.82
Man	512 × 512	28.9/0.91	30.5/0.93	30.3/0.93	31.6/0.96	25.0/0.74	25.0/0.73	25.3/0.78	26.5/0.86
Plane	256 × 256	40.0/0.95	40.3/0.95	40.8/0.95	40.7/0.95	34.1/0.91	32.6/0.90	32.2/0.90	35.3/0.92
Tank	512 × 512	32.0/0.90	32.7/0.92	32.7/0.92	34.2/0.95	29.4/0.79	29.5/0.79	29.7/0.81	31.0/0.88
Elaine	512 × 512	32.7/0.95	33.7/0.95	33.0/0.95	33.9/0.96	30.3/0.89	30.3/0.89	30.5/0.90	31.9/0.92
Clock	256 × 256	33.4/0.93	34.3/0.94	35.1/0.94	37.0/0.96	27.1/0.85	26.7/0.84	26.5/0.84	30.2/0.91

TABLE VI
PSNRs(dB)/SSIMs OF RECONSTRUCTED REMOTE SENSING IMAGES BY DIFFERENT COMPRESSION APPROACHES

Images	Resolution	Bit-rate = 0.4 bpp				Bit-rate = 0.1 bpp			
		MP	RLS-DLA	JPEG 2000	CCSR	MP	RLS-DLA	JPEG 2000	CCSR
Figure 12 (a)	420 × 420	30.0/0.89	30.5/0.90	31.2/0.93	30.6/0.96	24.7/0.69	24.1/0.66	24.5/0.69	25.7/0.84
Figure 12 (b)	512 × 512	24.5/0.84	24.9/0.86	24.8/0.87	26.1/0.90	22.0/0.65	21.9/0.65	22.1/0.67	22.7/0.72
Figure 12 (c)	256 × 256	29.7/0.79	30.1/0.82	30.6/0.83	32.2/0.91	25.9/0.63	25.6/0.58	25.7/0.61	27.4/0.76
Figure 12 (d)	512 × 512	28.7/0.90	29.0/0.91	28.8/0.91	30.1/0.93	25.6/0.75	25.3/0.73	25.6/0.76	26.7/0.84

we analyze such benefits through the experiment of compressing the *Lena* image.

Fig. 5 shows the reconstructed *Lena* images compressed by SR9 at 0.4 bpp and SR16 at 0.1 bpp, respectively. In this figure, the comparison is conducted on the proposed approach with and without considering the neighboring patch compatibility. Here, one pixel overlap was applied for the neighboring patch compatibility. As expected, reducing the block effect improves the visual quality of the reconstructed images with

little artifacts at boundaries, but slightly reduces the PSNRs of the reconstructed images, as there is a trade-off between the neighboring patch compatibility and the mean squared error.⁷ Consequently, one pixel overlap is sufficient for neighboring patch compatibility while effectively reducing block effect.

⁷ Along with the increased number of overlapping pixels, more patches have to be encoded. This results in more bits on encoding an input image, given the same reconstruction quality.

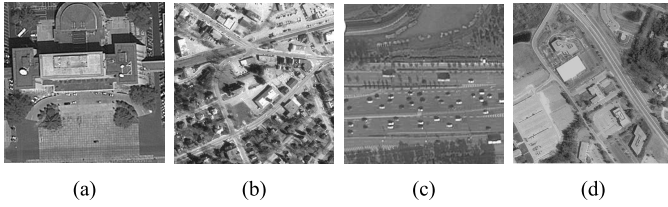


Fig. 12. Four original remote sensing images for compression. (a) Tsinghua Campus. (b) West Concord. (c) Road. (d) Concord (Sub-region).

Next, it is worth investigating the importance of coefficient compressibility on the rate-distortion performance of our CCSR approach. For the scenario of not enforcing coefficients compressible, we simply removed the constraint on compressibility by setting γ to 0 in (10). In addition, we applied our approach with decay rate $p = 1.67$ and 1, to analyze the impact of varying compressibility. Note that the over-complete dictionary was redesigned upon (16), with the same p and γ as (10) that are changed in this section. Fig. 6 shows the rate-distortion comparison of compressing the *Lena* image, using the proposed CCSR approach with varying coefficient compressibility and without coefficient compressibility. We can see from this figure that the coefficient compressibility demonstrates the advantage of the proposed approach in promoting the coding efficiency, especially at low bit-rates. However, the large compressibility (e.g., $p = 1$) can only achieve good rate-distortion performance at a specific low bit-rate, and is not well adaptive to the change of bit-rates. On the other hand, despite good adaptivity, small compressibility (e.g., $\gamma = 0$) cannot yield appropriate rate-distortion performance at low bit-rates. Therefore, in the CCSR approach there exists a tradeoff between adaptivity and low bit-rate rate-distortion performance. In this paper, we chose $p = 1.67$ to ensure that the CCSR approach works well at low bit-rates and is suitable for various bit-rates.

D. Evaluation of the Performance on Low Bit-Rate Image Compression

In this section, we concentrate on the performance evaluation of the proposed CCSR approach on compressing several natural images, including *Lena*, *Camera man*, *Rices*, *Ambulance*, as well as some remote sensing images.

First, in Fig. 7 we show the rate-distortion curves of the proposed and other conventional approaches on compressing the *Lena* image. The distortion in this figure was evaluated in terms of PSNR and structural similarity (SSIM) [31], from the aspects of both objective and subjective reconstruction quality. This figure demonstrates that the rate-distortion performance of our CCSR approach is superior to JPEG 2000, RLS-DLA, and MP approaches, at low bit-rates. We can further observe that if the bit-rate is >0.3 bpp, the performance of our CCSR approach with 9×9 patches (SR9) is better than that with 16×16 patches (SR16); otherwise SR16 outperforms SR9. Therefore, in the subsequent experiments, our CCSR approach utilized 9×9 patches at 0.4 bpp and 16×16 patches at 0.1 bpp.

Fig. 8 shows the visual quality of the *Lena* images compressed by different approaches at 0.1 bpp. From this

figure, it can be observed that the reconstructed image by JPEG 2000 has many blurring effects in some regions, and that the reconstructed image by RLS-DLA yields many jagged effects along the edges. Although MP produces more pleasant visual quality, it fails in encoding the texture details, e.g., the eyes and hairs. Compared with JPEG 2000, RLS-DLA, and MP, the proposed CCSR approach indeed provides better visual quality with more detailed texture and less blurring/jagged effects. However, the CCSR approach cannot totally eliminate the block effect, even the neighboring patch compatibility having been embedded. Other postprocessing deblocking algorithms, such as [32], may be applied to further deal with blocking artifacts of the CCSR approach. In a word, the proposed CCSR approach visually outperforms the JPEG 2000, RLS-DLA, and MP approaches, when compressing the *Lena* image at low bit-rates.

Also, we tested the proposed CCSR approach on other nine natural images, chosen from the Internet, MATLAB toolbox, and the SIPI database [30]. The compression bit-rates were set to 0.4 and 0.1 bpp, respectively. As discussed above, 9×9 patches were applied for image compression at 0.4 bpp (denoted by SR9), and 16×16 patches were utilized at 0.1 bpp (denoted by SR16). We report in Table V the reconstruction errors of the nine compressed images, evaluated in terms of PSNR and SSIM. Note that in Table V the images of *Plane*, *Tank*, *Elaine*, and *Clock* were chosen from the SIPI database. Fig. 9 shows the three original images of *Camera man*, *Rices*, and *Ambulance*. Beyond the results of Table V, Figs. 10 and 11 show the visual quality of the three images in Fig. 9, compressed by JPEG 2000, RLS-DLA, MP, and the CCSR approaches. From Figs. 9–11 and Table V, we can see that in comparison with JPEG 2000, RLS-DLA, and MP, our CCSR approach again offers the better performance on compressing most images at 0.4 bpp, in both visual and objective quality, and we can also see that for lower bit-rate (at 0.1 bpp) our CCSR approach provides much better visual and objective quality than the other three approaches, on compressing all images.

Now, we move to the compression results on the remote sensing images of Fig. 12. Figs. 13 and 14 show the reconstructed images of the proposed CCSR approach and other conventional approaches, at 0.1 or 0.4 bpp. As seen there, compared with the conventional JPEG 2000, RLS-DLA, and MP approaches, the CCSR approach yields more favorable visual quality with sharper edges and less blurred texture, especially at 0.1 bpp. However, it still suffers from the loss of small features. This is probably due to the fact that only high-valued coefficients of the CCSR approach are preserved at low bit-rates for the global features (e.g., object edges), and a large amount of low-valued coefficients reflecting detailed texture are discarded. Of course, the loss of detailed texture can be relieved by decreasing coefficient compressibility. As the cost, the global features, however, may be degraded heavily.

Table VI further tabulates the distortion of compressing the remote sensing images of Fig. 12, in the form of both PSNR and SSIM. Figs. 13 and 14 and Table VI clearly reveal that although there is no remote sensing image involved

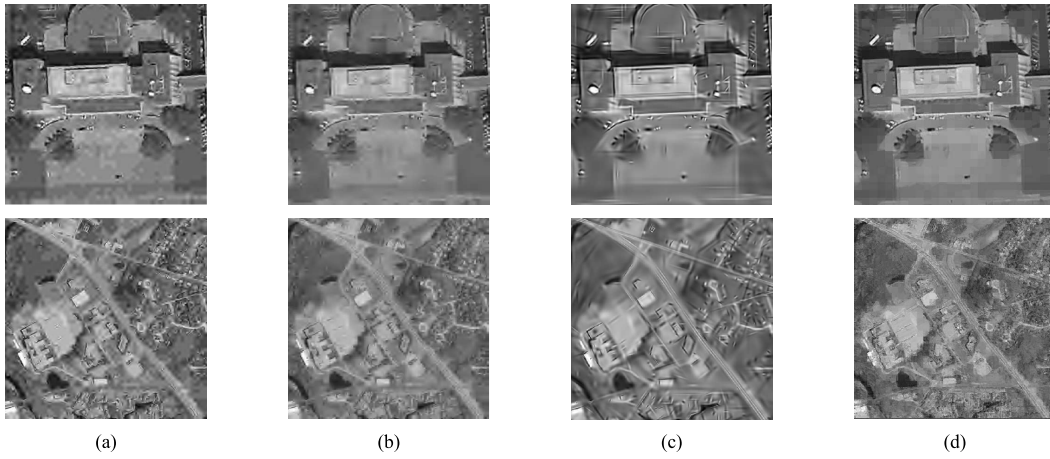


Fig. 13. Remote sensing images of Fig. 12(a) and (b) compressed by (a) RLS-DLA, (b) JPEG 2000, (c) MP, and (d) CCSR approaches, at 0.1 bpp.

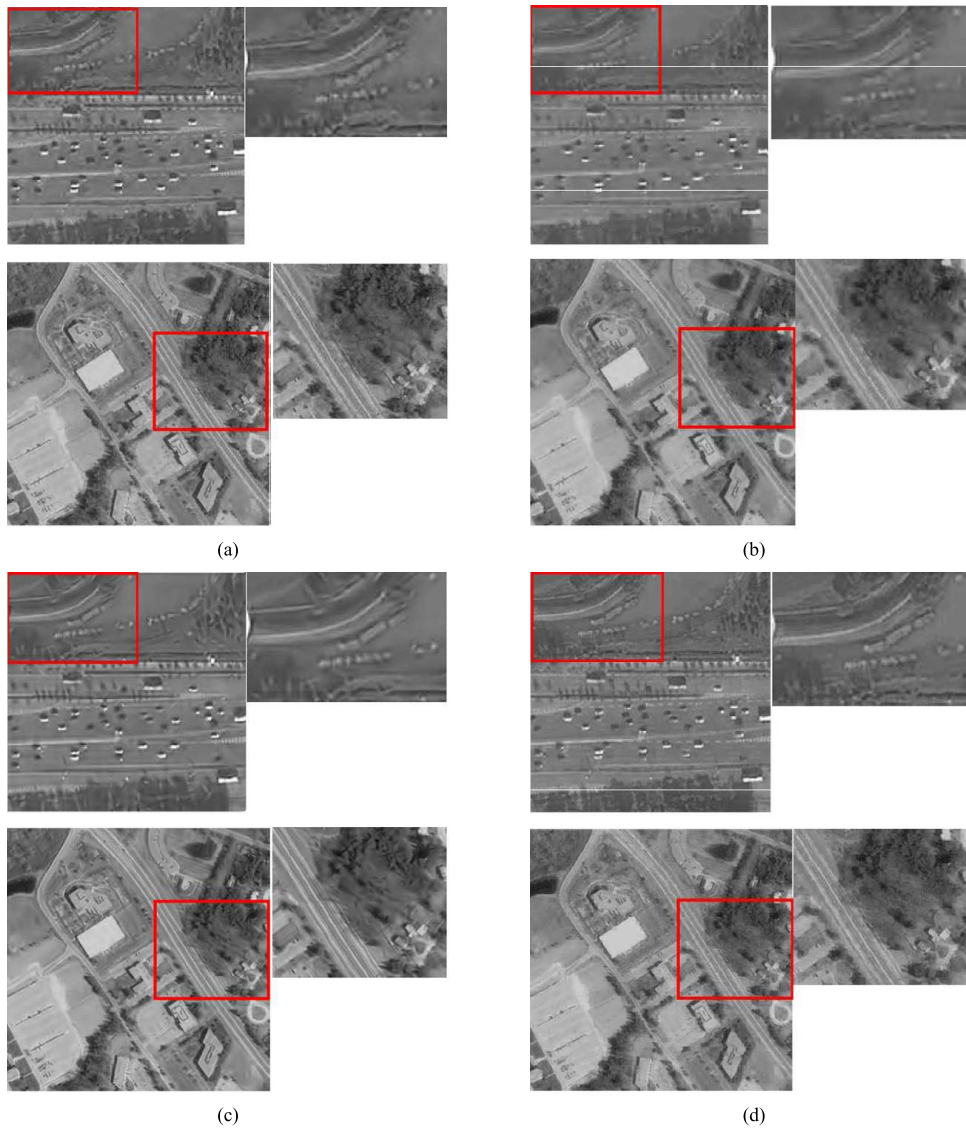


Fig. 14. Remote sensing images of Fig. 12(c) and (d) compressed by (a) RLS-DLA, (b) JPEG 2000, (c) MP, and (d) CCSR approaches, at 0.4 bpp.

in dictionary learning, the proposed CCSR approach is still capable of compressing the remote sensing images effectively at low bit-rates.

VI. CONCLUSION

In this paper, we have proposed a CCSR approach with learnt over-complete dictionary for compressing images at low

bit-rates. More specifically, in our CCSR approach, given the learnt over-complete dictionary, an image patch can be well represented with the linear combination of elements selected from this dictionary, based on the coefficients constrained by the proposed CCSR formulation. Then, a recursive algorithm was proposed to solve the linear optimization problem of the CCSR formulation, to obtain the sparse and compressible coefficients. Finally, the CCSR approach is capable of compressing images through the quantization and entropy encoding of compressible coefficients. Besides, the over-complete dictionary can be learnt off-line from a set of training image patches, using a gradient descent algorithm. The experimental results demonstrated that the proposed CCSR approach greatly outperforms the conventional JPEG 2000, RLS-DLA, and MP approaches in compressing images at low bit-rates.

Since the scope of this paper mainly concentrates on the sparse representation with learnt over-complete dictionary, it simply utilizes K-means method and Huffman coding as the scheme of quantization and entropy coding, for image compression. Other optimal quantization and entropy coding, for the proposed CCSR approach, is one of our future research directions. In addition, our CCSR approach can be further applied to video coding by considering the variational texture patches among the neighboring image frames. This can be considered as one of the promising future research directions. Also, the study on the robustness of our approach against corrupted data (e.g., lost coefficients) can be seen as another future direction.

APPENDIX PROOF OF (20)

We apply the matrix differentiation to derive the derivative formulation of (20). First, we expand (20) to

$$\begin{aligned} \sum_t \|y_t^* - D w_t^*\|_2^2 &= \sum_t \text{tr} \left((y_t^* - D w_t^*)(y_t^* - D w_t^*)^T \right) \\ &= \sum_t \text{tr}(y_t^* y_t^{*T}) - 2\text{tr}(D w_t^* y_t^{*T}) + \text{tr}(D w_t^* w_t^{*T} D^T). \end{aligned} \quad (22)$$

Since y_t^* as the training texture patch has already been known, the derivative of the above equation, with respect to D , can be written as

$$\begin{aligned} \frac{\partial}{\partial D} \sum_t \|y_t^* - D w_t^*\|_2^2 \\ = \frac{\partial}{\partial D} \sum_t \text{tr}(D w_t^* w_t^{*T} D^T) - \frac{\partial}{\partial D} \sum_t \text{tr}(2D w_t^* y_t^{*T}). \end{aligned} \quad (23)$$

Next, as $w_t^* w_t^{*T}$ is symmetric matrix, using the rules of derivative of matrix traces [33] we can obtain

$$\frac{\partial}{\partial D} \sum_t \text{tr}(D w_t^* w_t^{*T} D^T) = \sum_t 2D w_t^* w_t^{*T} \quad (24)$$

and

$$\frac{\partial}{\partial D} \sum_t \text{tr}(2D w_t^* y_t^{*T}) = \sum_t 2y_t^* w_t^{*T}. \quad (25)$$

Therefore, with w_t^* being fixed, (23) can be rewritten as

$$\frac{\partial}{\partial D} \sum_t \|y_t^* - D w_t^*\|_2^2 = 2 \sum_t (D w_t^* w_t^{*T} - y_t^* w_t^{*T}). \quad (26)$$

Finally, (20) is obtained.

REFERENCES

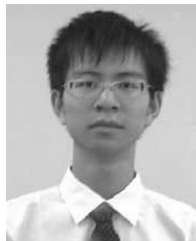
- [1] M. Xu, J. Lu, and W. Zhu, "Sparse representation of texture patches for low bit-rate image compression," in *Proc. IEEE VCIP*, Nov. 2012, pp. 1–6.
- [2] Y. Liu and B. Zalik, "An efficient chain code with Huffman coding," *Pattern Recognit.*, vol. 38, no. 4, pp. 553–557, 2005.
- [3] W. Dong, G. Shi, and J. Xu, "Adaptive nonseparable interpolation for image compression with directional wavelet transform," *IEEE Signal Process. Lett.*, vol. 15, no. 1, pp. 233–236, Jan. 2008.
- [4] G. K. Wallace, "The JPEG still picture compression standard," *Commun. ACM*, vol. 34, no. 4, pp. 607–609, 1991.
- [5] D. Taubman and M. Marcellin, *JPEG 2000: Image Compression Fundamentals, Standards and Practice*, 1st ed. Norwell, MA, USA: Kluwer, 2001.
- [6] S. G. Mallat and Z. F. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.
- [7] S. Chen and D. Donoho, "Basis pursuit," in *Proc. 28th Asilomar Conf. Signals, Syst. Computers*, Nov. 1994, pp. 41–44.
- [8] R. Neff and A. Zakhor, "Very low bit-rate video coding based on matching pursuits," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 1, pp. 158–171, Feb. 1997.
- [9] P. Frossard, P. Vanderghenst, R. M. Figueras i Ventura, and M. Kunt, "A posteriori quantization of progressive matching pursuit streams," *IEEE Trans. Signal Process.*, vol. 52, no. 2, pp. 525–535, Feb. 2004.
- [10] R. M. Figueras i Ventura, P. Vanderghenst, and P. Frossard, "Low-rate and flexible image coding with redundant representations," *IEEE Trans. Image Process.*, vol. 15, no. 3, pp. 726–739, Mar. 2006.
- [11] R. Rubinstein, A. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," *Proc. IEEE*, vol. 98, no. 6, pp. 1045–1057, Jun. 2010.
- [12] B. Olshausen and D. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6853, pp. 607–609, 1996.
- [13] J. Wright, M. Y. Ma, J. Mairal, G. Sapiro, T. Huang, and Y. Shuicheng, "Sparse representation for computer vision and pattern recognition," *Proc. IEEE*, vol. 98, no. 6, pp. 1031–1044, Jun. 2010.
- [14] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.
- [15] O. Bryta and M. Elad, "Compression of facial images using the K-SVD algorithm," *J. Vis. Commun. Image Represent.*, vol. 19, no. 4, pp. 270–282, 2008.
- [16] J. Zepeda, C. Guillemot, and E. Kijak, "Image compression using sparse representations and the iteration-tuned and aligned dictionary," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 5, pp. 1061–1073, Sep. 2011.
- [17] K. Skretting and K. Engan, "Image compression using learned dictionaries by RLS-DLA and compared with K-SVD," in *Proc. ICCASP*, May 2011, pp. 1517–1520.
- [18] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *J. Roy. Statist. Soc. B*, vol. 58, no. 1, pp. 267–288, 1996.
- [19] K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T. Lee, and T. J. Sejnowski, "Dictionary learning algorithms for sparse representation," *Neural Comput.*, vol. 15, no. 2, pp. 349–396, 2003.
- [20] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learn.*, vol. 11, no. 3, pp. 19–60, 2010.
- [21] J. Yang, J. Wright, T. Huang, and M. Yi, "Image super-resolution via sparse representation," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2861–2873, Nov. 2010.
- [22] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.
- [23] J. Yang, Z. Wang, Z. Lin, and S. Chen, "Coupled dictionary training for image super-resolution," *J. Mach. Learn.*, vol. 21, no. 8, pp. 3467–3478, Aug. 2012.

- [24] E. Candès and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.
- [25] H. S. Rauhut and K. P. Vandergheynst, "Compressed sensing and redundant dictionaries," *IEEE Trans. Inf. Theory*, vol. 54, no. 5, pp. 2210–2219, May 2008.
- [26] V. Cevher, "Learning with compressible priors," in *Advances in Neural Information Processing Systems*. Houston, TX, USA: Rice Univ., 2008.
- [27] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–136, Mar. 1982.
- [28] M. D. Adams. *The JasPer Project Home Page*. [Online]. Available: <http://www.ece.uvic.ca/~frodo/jasper/>, 2012.
- [29] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.
- [30] A. Weber. *The USC-SIPI Image Database*. [Online]. Available: <http://sipi.usc.edu/database/>, 2012.
- [31] W. Zhou, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
- [32] D. Sun and W.-K. Cham, "Postprocessing of low bit-rate block DCT coded images based on a fields of experts prior," *IEEE Trans. Image Process.*, vol. 16, no. 11, pp. 2743–2751, Nov. 2007.
- [33] J. R. Magnus and H. Neudecker, *Matrix Differential Calculus*. New York, NY, USA: Wiley, 1999.



Mai Xu (M'10) received the B.S. degree from Beihang University, Beijing, China, in 2003; the M.S. degree from Tsinghua University, Beijing, in 2006; and the Ph.D. degree from Imperial College London, London, U.K., in 2010.

He was a Research Fellow with the Electrical Engineering Department, Tsinghua University, from 2010 to 2012. Since 2013 he has been with Beihang University as an Associate Professor. He has authored more than 30 technical papers in international journals and conference proceedings. His research interests include visual communication and image processing.



Shengxi Li (S'14) is currently working toward the B.S. degree with Beihang University, Beijing, China.

His research interests include rate distortion theory and perceptual video coding.

Mr. Li received the Beihang Gold Medal Honor in 2013, which was the highest honor in Beihang University during his undergraduate study. He also received 15 scholarships, including the National Scholarship, and nearly 20 competition prizes.



Jianhua Lu (M'98–SM'07) received the B.S. and M.S. degrees from Tsinghua University, Beijing, China, in 1986 and 1989, respectively, and the Ph.D. degree in electrical and electronic engineering from Hong Kong University of Science and Technology, Hong Kong.

He has been with the Department of Electronic Engineering, Tsinghua University, since 1989, where he currently is a Professor. He has authored more than 180 technical papers in international journals and conference proceedings. His research interests

include broadband wireless communication, multimedia signal processing, and wireless networking.

Dr. Lu is a Senior Member of the IEEE Communication Society and the IEEE Signal Processing Society. He is currently a Chief Scientist of the National Basic Research Program, China. He has been an Active Member of professional societies. He has served in numerous IEEE conferences as a Technical Program Committee Member, and was a Lead Chair of the General Symposium of the IEEE International Conference on Communications in 2008, and a Program Committee Co-Chair of the 9th IEEE International Conference on Cognitive Informatics in 2010. He received the Best Paper Awards at the IEEE International Conference on Communications, Circuits and Systems in 2002; the Communications and Networking in China in 2006; and the IEEE Embedded-Com in 2012.



Wenwu Zhu (F'10) received the B.E. and M.E. degrees from National University of Science and Technology, Hefei, China, in 1985 and 1988, respectively; the M.S. degree from Illinois Institute of Technology, Chicago, IL, USA, in 1993; and the Ph.D. degree from Polytechnic Institute of New York University, Brooklyn, NY, USA, in 1996, all in electrical and computer engineering.

He was with Bell Laboratories, Shanghai, China, from 1996 to 1999. He was the Chief Scientist and the Director with Intel Research China, Beijing, from 2004 to 2008. He was also a Senior Researcher and Research Manager with Microsoft Research Asia, Beijing. He is a 1000 People Plan Professor of the Computer Science Department at Tsinghua University, Beijing, China. He has authored more than 200 refereed papers and holds 40 patents. His research interests include multimedia computing, communications and networking, including multimedia cloud computing, social media computing, and wireless multimedia communications.