SAR Image Compression Using Multiscale Dictionary Learning and Sparse Representation

Xin Zhan, Rong Zhang, Dong Yin, and Chengfu Huo

***Abstract*—In this letter, we focus on a new compression scheme for synthetic aperture radar (SAR) amplitude images. The last decade has seen a growing interest in the study of dictionary learning and sparse representation, which have been proved to perform well on natural image compression. Because of the special techniques of radar imaging, SAR images have some distinct properties when compared with natural images that can affect the design of a compression method. First, we introduce SAR properties, sparse representation, and dictionary learning theories. Second, we propose a novel SAR image compression scheme by using multiscale dictionaries. The experimental results carried out on amplitude SAR images reveal that, when compared with JPEG, JPEG2000, and a single-scale dictionary-based compression scheme, the proposed method is better for preserving the important features of SAR images with a competitive compression performance.**

***Index Terms*—Dictionary learning, image compression, sparse representation, synthetic aperture radar (SAR).**

I. Introduction

SYNTHETIC aperture radar (SAR) images formed from spatially overlapped radar phase histories are becoming increasingly important in a variety of remote sensing applications. Compression is necessary because of the requirements for transmitting high-volume data at real time and storing them efficiently on the ground station. State-of-the-art techniques are divided into two groups: compression of the SAR raw data (complex numbers) [1], [2] and compression of the synthesized amplitude images [3]–[5]. In this letter, we focus on the compression of amplitude SAR images.

Some special characteristics of SAR images affect the design of a compression scheme. The first is the speckle phenomenon resulting from the coherent radiation and processing. Second, both very detailed texture and large homogeneous regions can be found in SAR images. These SAR properties will be described in more detail in Section II. Most image compression algorithms rely on appropriate signal representation, like the discrete cosine transform for JPEG or the discrete wavelet transform for JPEG2000. Recent years have witnessed a growing interest in the research for sparse representation of signals [6]. Instead of using a fixed transform based on a mathematical model, an overcomplete dictionary is learned from a training set, and sparse coding is applied to decompose the signal into a linear combination of a few atoms from the dictionary. These new theories have successfully been used for image compression [7], [8]. However, most of them are designed for natural images, and the learned dictionary is only of single scale. In this letter, an analysis of making dictionary learning more adapted to the preservation of SAR features is presented, and a multiscale approach is proposed to improve the compression performance.

The rest of this letter is organized as follows. In Section II, we introduce SAR properties, sparse coding, and dictionary learning. Section III describes the proposed multiscale compression scheme. The experimental results are presented in Section IV, and this letter concludes in Section V.

II. SAR Properties and Sparse Representation

A. SAR Properties

Like all coherent imaging systems, SAR images suffer degradation owing to speckle noise. Researchers have established good statistical models to describe the speckle phenomenon [9]. In the widely used multiplicative model, the pixel amplitude Y is approximated by the product of two independent random variables

$$Y = XS \tag{1}$$

with X denoting the radar reflectivity (speckle-free magnitude) and S denoting the speckle noise with mean

$$E(S) = \frac{\Gamma(L+1/2)}{L^{\frac{1}{2}}\Gamma(L)} \tag{2}$$

and variance

$$\mathrm{Var}(S) = 1 - E^2(S) \tag{3}$$

assuming that the amplitude format SAR images are formed by averaging L independent intensity samples followed by taking a square-root operation.

Texture measure based on the multiplicative model is simple but effective. Texture in SAR images is a combination of two components [4]: the intrinsic texture of the scene backscatter and the fluctuations owing to speckle. After some manipulation

Manuscript received May 24, 2012; revised September 28, 2012, October 31, 2012, and November 11, 2012; accepted November 12, 2012. Date of publication January 23, 2013; date of current version June 13, 2013. This work was supported in part by the National Grand Fundamental Research 973 Program of China under Grant 2010CB731904 and in part by the National Nature Science Foundation of China under Grant 61172154.

The authors are with the Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei 230027, China (e-mail: zhanxin@mail.ustc.edu.cn; zrong@ustc.edu.cn; yindong@ustc.edu.cn; roy@mail.ustc.edu.cn).

Digital Object Identifier 10.1109/LGRS.2012.2230394

of (1), the variance of the normalized intrinsic texture $\mathrm{Var}(T_X)$ is represented as

$$\begin{aligned}\mathrm{Var}(T_X) &= \frac{\mathrm{Var}(X)}{E^2(X)} \\ &= \frac{\mathrm{Var}(Y)}{E^2(Y)}\left(E^2(S) - \mathrm{Var}(s)\frac{E^2(Y)}{\mathrm{Var}(Y)}\right) \\ &= \mathrm{Var}(T_Y) * \gamma. \end{aligned} \tag{4}$$

Here, $E(Y)$ is the local mean of Y, and $\mathrm{Var}(Y)$ is the corresponding variance, which can be estimated from a small window around the pixel. On heterogeneous areas, $\mathrm{Var}(T_Y) \simeq \mathrm{Var}(T_X)$, and thus, the texture factor $\gamma \simeq 1$. On the other hand, when the area is homogeneous, $\mathrm{Var}(T_Y) \gg \mathrm{Var}(T_X)$, and thus, $\gamma \simeq 0$. In our experiments, a pixel block is considered as representing a textured (otherwise textureless) area if $\gamma > \mathrm{Var}(T_S)/\mathrm{Var}(T_Y)$.

B. Sparse Coding

Suppose that we have a signal $\boldsymbol{b} \in \mathbb{R}^N$ and a dictionary whose elements, called atoms, are stored as columns of a matrix $\boldsymbol{D} \in \mathbb{R}^{N\times K}$. The dictionary $\boldsymbol{D}$ is overcomplete when $K > N$, implying that

$$\boldsymbol{b} = \boldsymbol{D}\boldsymbol{w} \tag{5}$$

lacks a unique solution. Sparse coding aims at seeking approximations of $\boldsymbol{b}$ with as few atoms in $\boldsymbol{D}$ as possible by solving the minimization problem [6]

$$\min_{\boldsymbol{w}} \|\boldsymbol{w}\|_0, \qquad \text{subject to } \|\boldsymbol{b} - \boldsymbol{D}\boldsymbol{w}\|_2^2 \le N\sigma^2. \tag{6}$$

We use $\|\cdot\|_0$ to denote the ℓ_0 pseudonorm, which is the number of nonzero entries of a vector. The resulted $\boldsymbol{w}$ of (6) gives the smallest set of atoms in $\boldsymbol{D}$ necessary to approximate $\boldsymbol{b}$ within the noise level specified by the noise variance σ^2.

Such a problem is known to be NP-hard, and many algorithms have been developed to find $\boldsymbol{w}$ efficiently, including orthogonal matching pursuit (OMP), basis pursuit, and focal underdetermined system solver [10]. In this letter, OMP is used because of its approximation capability and computational complexity. In each iteration step of OMP, one atom that reduces the residual most is selected. This process is repeated until the residual reaches the noise level.

C. Dictionary Learning

Compared with orthogonal transforms, the learned dictionary is well adapted to its purpose, i.e., spare representation of a specific class of signals. Given a training set $\boldsymbol{B} = \{\boldsymbol{b}_i \in \mathbb{R}^N\}_{i=1}^M$, we need to find the dictionary $\boldsymbol{D} \in \mathbb{R}^{N\times K}$ by solving the following optimization problem:

$$\min_{\boldsymbol{D},\boldsymbol{W}} \|\boldsymbol{W}\|_0, \qquad \text{subject to } \|\boldsymbol{B} - \boldsymbol{D}\boldsymbol{W}\|_F^2 \le MN\epsilon^2 \tag{7}$$

where $\boldsymbol{W} = \{\boldsymbol{w}_i \in \mathbb{R}^K\}_{i=1}^M$ is the corresponding coefficient set and $\|\cdot\|_F$ denotes the Frobenius norm. The training limit ϵ^2 is used as the stop criterion.

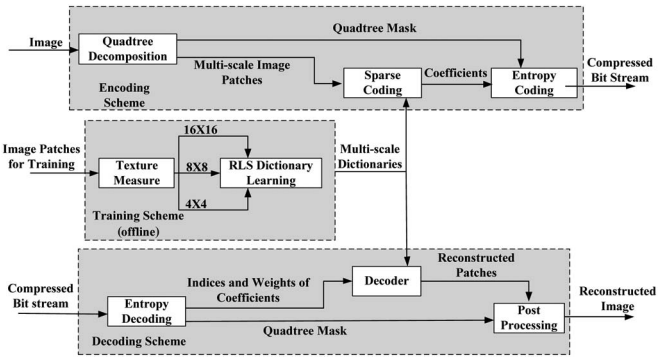

Fig. 1. Block diagram of the proposed compression scheme.

The dictionary learning algorithms that have been proposed so far share a common two-step process.

1) Find the sparse coefficients given the dictionary.
2) Update the dictionary assuming that the sparse vectors are fixed.

Their main differences rely on the methods used to successively estimate the sparse vectors and the dictionary. Method of optimal directions [6] updates the dictionary by global least squares fit, and the K-singular value decomposition algorithm [8] relies on singular value decomposition. The recursive least squares dictionary learning algorithm (RLS-DLA) provided in [11], which is adopted in this work, achieves the best performance. More details can be found in [11].

III. Proposed Multiscale Compression Scheme

The block diagram of the proposed compression scheme, as shown in Fig. 1, consists of three parts: the training, encoding, and decoding schemes. First, multiscale dictionaries are obtained through the training scheme. Note that these dictionaries are not included in the compressed bit stream as they are trained offline. Second, the SAR image to be compressed is sliced into nonoverlapping patches of different sizes based on quadtree decomposition. These patches are sparse coded by using the corresponding dictionary, and finally, the sparse coding coefficients are quantized and entropy coded. The detailed descriptions are given in the following.

A. Multiscale Dictionary Learning

Multiscale dictionaries $\boldsymbol{D}_{i=1,2,3}$ are learned through the training scheme. It is formally presented in Algorithm 1. Texture measure based on the multiplicative model of SAR (see Section II-A) is used as the homogeneity criterion, which divides the training patches into three-stage blocks of sizes 16×16, 8×8, and 4×4 successively. It seems better to do this decomposition using a cost function defined as the coding bits as well as the distortion, but these cannot be estimated before the dictionary is learned. In this letter, we try to choose an alternative cost function with low computational complexity. All of the three values, i.e., the mean-squared value, the maximum absolute value, and the texture measure, are used as criteria. It is found that the mean-squared value and the maximum absolute value are not robust for speckles and that the texture measure

using the SAR image model, which is employed here, gives the best results. After decomposition, most of the homogeneous areas appear in the 16×16 patches, and the edges, detailed textures, and point targets are divided further. Then, these training patches are dc components removed and presented to the RLS-DLA (see Section II-C). Parameters $k_{i=1,2,3}$ decide the number of atoms of each dictionary. In our experiments, different $k_{i=1,2,3}$'s are tested and finally adjusted to 512, 256, and 64, respectively.

Algorithm 1 Multiscale Dictionary Learning

INPUT:

The training images $\boldsymbol{I}_t$, the atom numbers $k_{i=1,2,3}$.

OUTPUT:

Three-scale training sets $\boldsymbol{B}_1 = \{\boldsymbol{b}_1 \in \mathbb{R}^{16\times16}\}$, $\boldsymbol{B}_2 = \{\boldsymbol{b}_2 \in \mathbb{R}^{8\times8}\}$, and $\boldsymbol{B}_3 = \{\boldsymbol{b}_3 \in \mathbb{R}^{4\times4}\}$. Multiscale dictionaries $\boldsymbol{D}_1 \in \mathbb{R}^{256\times k_1}$, $\boldsymbol{D}_2 \in \mathbb{R}^{64\times k_2}$, and $\boldsymbol{D}_3 \in \mathbb{R}^{16\times k_3}$.

PROCEDURE:

- Training Sets
  1) Initialize $\boldsymbol{B}_{i=1,2,3} = \phi$.
  2) Calculate the normalized speckle texture $\mathrm{Var}(T_S)$ by using (2) and (3).
  3) Extract overlapping 16×16 patch set $\boldsymbol{P}_1$ from $\boldsymbol{I}_t$.
  4) For each patch $\boldsymbol{P}_{1j}$
     Calculate the texture factor $\gamma_{\boldsymbol{P}_{1j}}$ by using (4).
     If $\gamma_{\boldsymbol{P}_{1j}} > \mathrm{Var}(T_S)/\mathrm{Var}(T_{\boldsymbol{P}_{1j}})$ (Texture Measure.)
     Divide it into 8×8 patches $\boldsymbol{P}_2$.
     else $\boldsymbol{B}_1 \leftarrow \boldsymbol{B}_1 \cup \boldsymbol{P}_{1j}$.
  5) For each patch $\boldsymbol{P}_{2j}$
     If $\gamma_{\boldsymbol{P}_{2j}} > \mathrm{Var}(T_S)/\mathrm{Var}(T_{\boldsymbol{P}_{2j}})$
     Divide it into 4×4 patches $\boldsymbol{P}_3$ and $\boldsymbol{B}_3 \leftarrow \boldsymbol{B}_3 \cup \boldsymbol{P}_{3j}$.
     else $\boldsymbol{B}_2 \leftarrow \boldsymbol{B}_2 \cup \boldsymbol{P}_{2j}$.
- Dictionaries
  1) Normalize the vectors of $\boldsymbol{B}_{i=1,2,3}$ and remove the dc components.
  2) Initialize $\boldsymbol{D}_i$ by using the k_i random vectors of $\boldsymbol{B}_i$, $i = 1, 2, 3$.
  3) Train the dictionaries:

$$\boldsymbol{D}_i = \mathbf{RLS\_DLA}(\boldsymbol{B}_i, k_i), \qquad i = 1, 2, 3.$$

(See Section II-C.)

B. Sparse Decomposition

The image $\boldsymbol{I}$ to be compressed is sparse decomposed over the learned multiscale dictionaries $\boldsymbol{D}_{i=1,2,3}$, as described in Algorithm 2. Quadtree decomposition is used to slice the image into nonoverlapping patches with the same homogeneity criterion as in Algorithm 1. Fig. 2 shows the quadtree structure for one of the test images depicting the final patch sizes used for different regions. In order to recover the image in the decoder, the quadtree mask is arithmetic encoded and translated as side information. Then, the different scale patches are dc components extracted and sparse coded separately by using the corresponding scale dictionary. The OMP algorithm is applied to solve the vector selection problem. After the sparse decomposition of the image $\boldsymbol{I}$, the dc components, the set of selected atom indices $\boldsymbol{S}_{i=1,2,3}$, and nonzero approximation weights $\boldsymbol{W}_{i=1,2,3}$ of the ac components are found and are then quantized and entropy coded further.

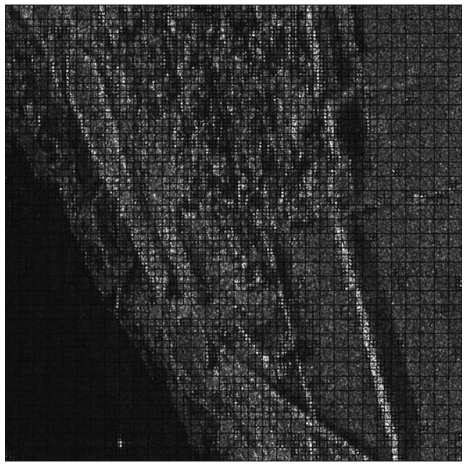

Fig. 2. Quadtree decomposition of an amplitude E-SAR image using texture measure. The original image is shown in Fig. 3(c) in Section IV.

C. Quantization and Coding

The dc components are coded separately by using a uniform quantizer and differential pulse code modulation (DPCM) predictor followed by an adaptive arithmetic coder to encode the prediction residuals. The ac components are made into two sequences, one for position indices $\boldsymbol{S}_{i=1,2,3}$ and the other for nonzero values $\boldsymbol{W}_{i=1,2,3}$. The position indices $\boldsymbol{S}_{i=1,2,3}$ are DPCM predicted and arithmetic encoded. The approximation weights $\boldsymbol{W}_{i=1,2,3}$ are uniformly quantized using quantization steps $\Delta_{i=1,2,3} = \Delta * \gamma_{i=1,2,3}$, where $\gamma_{i=1,2,3}$ denotes the average texture factors for the three-scale patches and Δ is the basic quantization step that decides the final compression bit rate. This adaptive quantization is equivalent to an efficient bit allocation approach, and more bits will be allocated to smaller textured patches in order to preserve the important features for SAR data. Finally, the quantized $\boldsymbol{W}_{i=1,2,3}$'s are entropy coded into bit streams directly. Compared with the sophisticated coding strategies designed for some other compression schemes (JPEG and JPEG2000), our coding strategy is simple, and we plan to improve it in future works.

Algorithm 2 Multiscale Sparse Decomposition of Image

INPUT:

The image $\boldsymbol{I}$ and the multiscale dictionaries $\boldsymbol{D}_{i=1,2,3}$.

OUTPUT:

The dc components, the set of selected atom indices $\boldsymbol{S}_{i=1,2,3}$, and the approximation weights $\boldsymbol{W}_{i=1,2,3}$.

PROCEDURE:

1) Initialize $\boldsymbol{S}_{i=1,2,3} = \phi$ and $\boldsymbol{W}_{i=1,2,3} = \phi$.

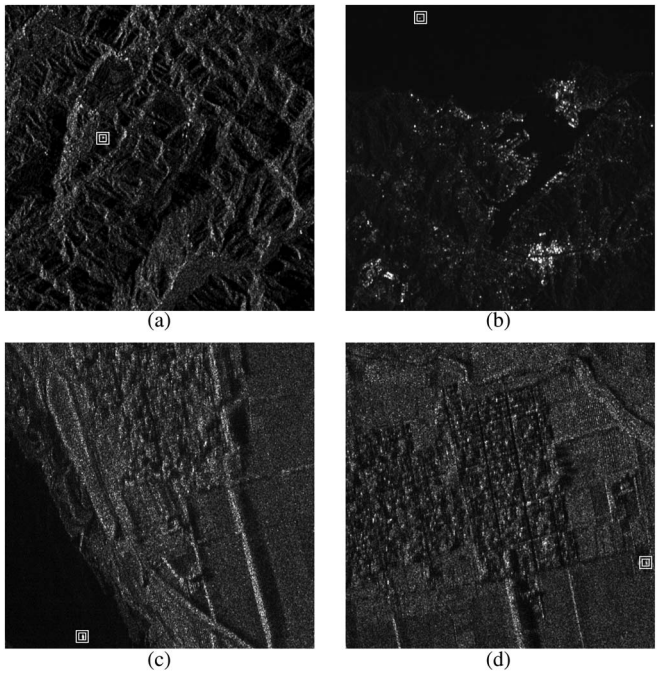

Fig. 3. Four test images. (a) and (b) Amplitude RADARSAT-2 data (April 16, 2008, Vancouver, Canada). (c) and (d) Amplitude E-SAR data (November 17, 2004, Balikpapan, Kalimantan, Indonesia).

2) Quadtree decomposition of $\boldsymbol{I}$ by using the same texture measure in Algorithm 1. Nonoverlapping patch sets $\boldsymbol{P}_1 \in \mathbb{R}^{16\times 16}$, $\boldsymbol{P}_2 \in \mathbb{R}^{8\times 8}$, and $\boldsymbol{P}_3 \in \mathbb{R}^{4\times 4}$ are obtained.
3) Calculate the average texture factor $\gamma_{i=1,2,3}$ for $\boldsymbol{P}_{i=1,2,3}$, which will be used for adaptive quantization in Section III-C.
4) Extract the dc components of $\boldsymbol{P}_{i=1,2,3}$.
5) Sparse coding

$$\{\boldsymbol{S}_i, \boldsymbol{W}_i\} = \mathbf{OMP}(\boldsymbol{P}_i, \boldsymbol{D}_i), \qquad i = 1, 2, 3.$$

(See Section II-B.)

IV. Results

For our experiments, the method was tested on RADARSAT-2 spaceborne and E-SAR airborne data, each 512×512 pixels large and represented with 16 bpp as shown in Fig. 3(a)–(d). These test images contain both the natural scenes (mountains and forests) and manmade scenes (cities and villages) that typically appear in SAR images. In order to assess the effectiveness of the proposed scheme, four other methods were applied: JPEG, JPEG2000, an approach using single-scale dictionary [8], and the online training method. The distortion is measured by signal-to-noise ratio (SNR)

$$\text{SNR} = 10\ \log_{10}\left(\frac{\sigma_x^2}{\sigma_e^2}\right) \tag{8}$$

where σ_x^2 denotes the variance of the original image and σ_e^2 denotes the variance of error between the original and reconstructed data. The experimental results with various compression bit rates are listed in Table I, and Table II specifies the coding cost of different parts of the proposed scheme on image (c) at 3 bpp in particular. For the results of the online training method, the storage of the dictionaries is not included in the reported bit rate. This method is just applied to measure the performance advantage of using online-trained dictionaries over offline-trained dictionaries.

TABLE I
SNR (in Decibels) for Compressed SAR Images

Bit rate	1bpp	2bpp	3bpp	4bpp	8bpp
Image ID			(a)		
JPEG	X	20.61	25.06	29.45	52.77
Single RLS-DLA	16.84	22.56	27.92	33.38	58.04
JPEG2000	17.1	23.05	28.51	33.98	57.51
Proposed Scheme	17.27	23.18	28.67	34.18	58.87
On-line Training	17.35	23.4	28.91	34.52	59.29
Image ID			(b)		
JPEG	X	25.03	28.45	33.47	55.01
Single RLS-DLA	19.56	26.14	32.12	37.49	61.98
JPEG2000	19.78	27.02	32.73	38.32	61.48
Proposed Scheme	19.96	27.23	32.99	38.51	62.55
On-line Training	20.07	27.47	33.26	38.92	63.19
Image ID			(c)		
JPEG	X	X	23.85	26.22	48.92
Single RLS-DLA	13.94	19.68	25.41	31.16	55.56
JPEG2000	14.23	19.85	25.89	31.45	54.19
Proposed Scheme	14.36	20.06	26.05	31.7	55.51
On-line Training	14.42	20.21	26.23	31.93	56.32
Image ID			(d)		
JPEG	X	X	23.05	25.32	47.57
Single RLS-DLA	13.08	17.92	23.61	29.37	53.86
JPEG2000	13.12	18.04	23.88	29.52	52.45
Proposed Scheme	13.39	18.29	23.9	29.83	54.17
On-line Training	13.45	18.42	24.15	30.18	54.9

TABLE II
Coding Cost of the Proposed Scheme on Image (c) at 3 bpp

		Coding bits
Quadtree mask		4002
16X16 patches	DC components	2467
	position indices	76517
	coefficient weights	160351
8X8patches	DC components	11878
	position indices	93992
	coefficient weights	276922
4X4 patches	DC components	14536
	position indices	33841
	coefficient weights	117379

JPEG will no longer work at some low bit rates. The results show that the proposed scheme outperforms JPEG2000 by about 0.2 dB and that it is much better than JPEG. We notice that JPEG2000 benefits a lot from the embedded block coding with optimized truncation strategy, whereas ours is relatively simple. The performance of the proposed method can be improved further if more sophisticated quantization and entropy-coding mechanisms are applied. It is also interesting that, when the image is high-quality lossy compressed, the dictionary-learning-based approaches have advantages that are more obvious. Because the dictionaries are learned to approximate the original training images with little loss, they would be more powerful when the reconstructed images become closer to the original ones. We notice that the online method is indeed better than the offline approach. However, the online trained dictionaries should be translated losslessly along with the coefficient bit streams, and this coding

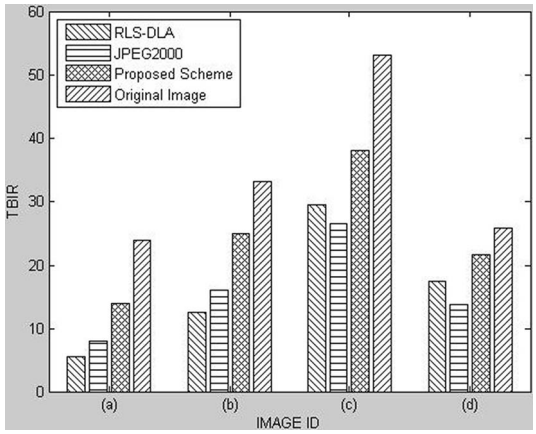

Fig. 4. Effects of compression on target preserving measure TBIR at 2 bpp.

overhead is too high. For example, 9 502 720 b is needed to store the three-scale dictionaries that are single-precision matrixes, and they are nearly ten times as great as the total number of the bits listed in Table II. Moreover, the heavy computational burden is another reason that makes online training unfeasible for real-time applications.

Taking into account the special characteristics of SAR, SNR only is not enough to evaluate the compression performance. Here, the ability of preserving specified targets of interest is also used as a quality measure. Several targets are selected from the test images with a target window and a background window. For each original and reconstructed image, target-to-background interference ratio (TBIR) [12] is calculated as

$$\text{TBIR} = \frac{\overline{I}_T - \overline{I}_B}{\sqrt{\sigma_T \sigma_B}} \tag{9}$$

where $\overline{I}_T$ is the mean of the target window, $\overline{I}_B$ is the mean of the background region, and σ_T and σ_B are the standard deviations of these regions. It measures how the targets stand out of the surrounding clutter, and a higher TBIR means better target preservation. The TBIR results obtained by using different methods at 2 bpp are shown in Fig. 4. It has been proved that sparse representation and dictionary learning are effective in automatic target recognition and classification for SAR images, and our compression results also show their potential in target characterization and preservation.

The computational complexity of the dictionary learning scheme is high because dozens of iterations are required. However, the dictionaries are trained offline with a representative training set so that it has no effect on the encoding and decoding running times. The running time is mainly spent in the sparse coding of the test image. In the experiments, we use the sparse modeling software [13] for sparse coding, which has an extremely fast implementation of the OMP algorithm by the Cholesky decomposition. The quadtree decomposition is implemented using Matlab code, and the others are written in C++. All algorithms are tested on a regular PC (Pentium Dual-Core CPU; 2.70 GHz; 2-GB RAM). For a 16-bpp test image, 512×512 [Fig. 3(c)], the running times are as follows: 0.156 s for quadtree decomposition, 0.109 s for sparse coding, and 0.160 s for quantization and entropy coding. The total encoding time of the proposed scheme is 0.425 s. Meanwhile, JPEG takes 0.29 s, and JPEG2000 takes 0.23 s. The execution time for the proposed decoder is 0.14 s.

V. Conclusion

In this letter, we have designed a novel SAR image compressor based on multiscale dictionary learning. The experimental results have revealed that the proposed scheme outperforms the still image compression standards JPEG and JPEG2000 and that it is more powerful in capturing the important structural features and targets of SAR images. This method can be improved further in two aspects: 1) Quantize and code the coefficients adaptively, and 2) make the trained dictionary shape adaptive. Our experiments prove that dictionary learning and sparse representation theories have great potential in SAR image compression and interpretation applications. It is hoped that this letter will motivate future research works in this field.

Acknowledgment

The authors would like to thank the reviewers and editors for their constructive suggestions and valuable comments.

References

[1] T. Ikuma, M. Naraghi-Pour, and T. Lewis, "Predictive quantization of range-focused SAR raw data," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 4, pp. 1340–1348, Apr. 2012.
[2] E. Magli and G. Olmo, "Lossy predictive coding of SAR raw data," *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 5, pp. 977–987, May 2003.
[3] D. Gleich, B. Gergic, Z. Cucej, and P. Planinsic, "Fuzzy-coded space-frequency quantization for SAR data compression," *IEEE Geosci. Remote Sens. Lett.*, vol. 1, no. 2, pp. 90–93, Apr. 2004.
[4] G. Mercier, "Reflectivity estimation for SAR image compression," *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 4, pp. 901–906, Apr. 2003.
[5] Z. Zeng and I. G. Cumming, "SAR image data compression using a tree-structured wavelet transform," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 3, pp. 546–552, Mar. 2001.
[6] M. Elad, *Sparse and Redundant Representations, From Theory to Applications in Signal and Image Processing*. New York: Springer-Verlag, 2010.
[7] A. Rahmoune, P. Vandergheynst, and P. Frossard, "Sparse approximation using M-term pursuit and application in image and video coding," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1950–1962, Apr. 2012.
[8] K. Skretting and K. Engan, "Image compression using learned dictionaries by RLS-DLA and compared with K-SVD," in *Proc. IEEE ICASSP*, May 2011, pp. 1517–1520.
[9] H. Xie, L. E. Pierce, and F. T. Ulaby, "Statistical properties of logarithmically transformed speckle," *IEEE Trans. Geosci. Remote Sens.*, vol. 40, no. 3, pp. 721–727, Mar. 2002.
[10] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2231–2242, Oct. 2004.
[11] K. Skretting and K. Engan, "Recursive least squares dictionary learning algorithm," *IEEE Trans. Signal Process.*, vol. 58, no. 4, pp. 2121–2130, Apr. 2010.
[12] M. Dutkiewicz and I. Cumming, "Evaluation of the effects of encoding on SAR data," *Photogramm. Eng. Remote Sens.*, vol. 60, no. 7, pp. 895–904, Jul. 1994.
[13] Sparse Modeling Software. [Online]. Available: http://spams-devel.gforge.inria.fr