Hindawi
International Journal of Digital Multimedia Broadcasting
Volume 2017, Article ID 3902543, 10 pages
https://doi.org/10.1155/2017/3902543

Research Article

Adaptive Image Compressive Sensing Using Texture Contrast

Fang Sun,[1] Dongyue Xiao,[2] Wei He,[1] and Ran Li[1,3]

[1]*School of Computer and Information Technology, Xinyang Normal University, Xinyang 464000, China*
[2]*School of Electrical and Electronic Engineering, Nanyang Institute of Technology, Nanyang 473000, China*
[3]*School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210003, China*

Correspondence should be addressed to Ran Li; liran358@163.com

Received 20 October 2016; Revised 16 February 2017; Accepted 1 March 2017; Published 13 March 2017

Academic Editor: Jintao Wang

The traditional image Compressive Sensing (CS) conducts block-wise sampling with the same sampling rate. However, some blocking artifacts often occur due to the varying block sparsity, leading to a low rate-distortion performance. To suppress these blocking artifacts, we propose to adaptively sample each block according to texture features in this paper. With the maximum gradient in 8-connected region of each pixel, we measure the texture variation of each pixel and then compute the texture contrast of each block. According to the distribution of texture contrast, we adaptively set the sampling rate of each block and finally build an image reconstruction model using these block texture contrasts. Experimental results show that our adaptive sampling scheme improves the rate-distortion performance of image CS compared with the existing adaptive schemes and the reconstructed images by our method achieve better visual quality.

1. Introduction

The core of traditional image coding (e.g., JPEG) is the image transformation based on Nyquist sampling theorem. It can recover an image without distortion only when the transformation number is greater than or equal to the total pixel number of image. However, limited by computation capability, the wireless sensor cannot tolerate excessive transformations, so traditional image coding is not fit for the wireless sensor with a light load [1, 2]. Besides, owing to the information focus on a few transformation coefficients, the quality of reconstructed image deteriorates greatly once several important coefficients are lost. Recently, the rapid development of Compressive Sensing (CS) [3, 4] introduces a new way to solve these defects in traditional image coding. Breaking the limitation of Nyquist sampling rate, CS accurately recovers signals using partial transformations. The superiority of CS lies in the fact that it can compress image by dimensionality reduction while transforming image, which attracts lots of researchers to develop the CS-based low-complexity coding [5, 6].

Many scholars are devoted to improving rate-distortion performance of image CS. A popular method is adopted to construct a sparse representation model to improve the convergence performance of minimum l_1-norm recovery; for example, Chen et al. [7] predict sparse residual using multihypothesis prediction; Becker et al. [8] exploit the first-order Nesterov's method to perform efficient sparse decomposition; Zhang et al. [9] use both local sparsity and nonlocal self-similarity to represent natural images; Yang et al. [10] use Gaussian mixture model to generate sparser representation. From different perspectives, these sparse representation schemes achieve some improvement of rate-distortion performance. However, their disadvantage is that rapid increase of computational complexity in spatial resolution, for example, the proposed algorithm by Zhang et al. [9], requires about an hour to recover an image of 512 × 512 in size. To avoid the high computational complexity, some works try to improve the quantization performance according to the statistics of CS samples. An efficient quantizer can reduce the amount of bits; for example, Wang et al. [11] exploit the hidden correlations between CS samples to design progressive fixed-rate scalar quantization; Mun and Fowler [12] and Zhang et al. [13] use Differential Pulse-Code Modulation (DPCM) to remove the redundancy between block CS samples. By reducing statistical redundancies, these quantizers obtain some performance improvements with a

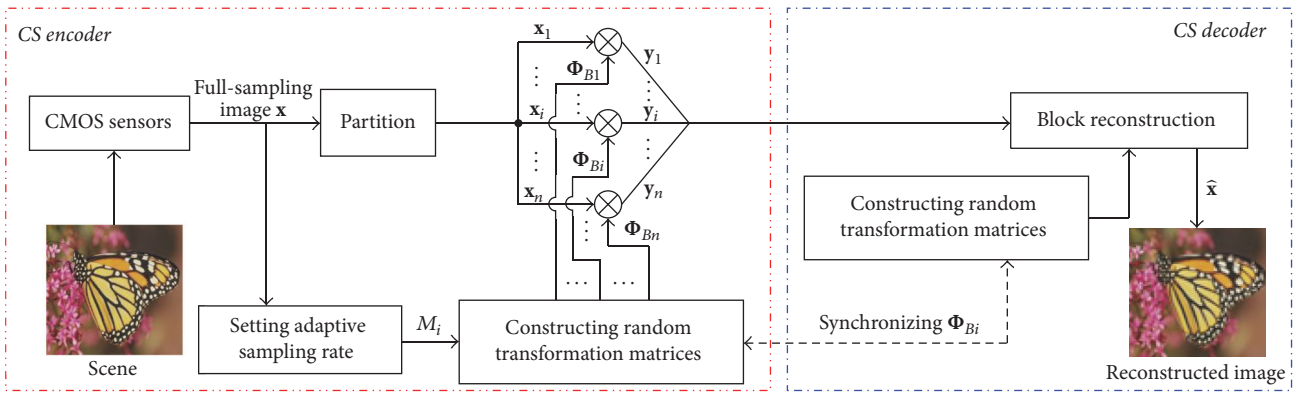


Figure 1: Framework of adaptive block CS.

low computational complexity. Despite its less computational burden, the quantization scheme has limited improvement of rate-distortion performance due to the lower redundancies in CS samples [14]. From the above, we can see that there is a tradeoff between computational complexity and quality improvement for image CS. We expect to find a scheme which strikes a balance between the two. Compared with sparse representation and quantization schemes, the feature-based adaptive sampling achieves a satisfying improvement of rate-distortion performance without introducing excessive computations. Its idea is to increase the efficiency of CS sampling by suppressing useless CS samples. The sampling rate of each block is allocated according to various image features; for example, Zhang et al. [15] determine the sampling rate of each block depending on varying block variance; Canh et al. [16] exploit the edge information to adaptively assign the sampling rate for each block. Block variance and edge information mean a low-level vision. They preserve the low-frequency information but neglect the high-frequency texture details attractive to human eyes. Oriented by the two-feature measures, lots of CS samples are invested into those blocks with simple patterns, which results in an undesirable reconstruction quality. To overcome the defect of traditional sampling scheme, useful features should be extracted to express high-level vision. Directed by interesting features, an efficient adaptive scheme can guarantee the recovery of high-frequency details.

Texture as a visual feature is used to reveal similar patterns independent of color and brightness, and it is the mutual inherent property existing in object surfaces; for example, tree, cloud, and fabric have their own texture details. Texture details contain important information on structures of object surfaces, revealing relations between object and its surrounds. Texture details represent high-frequency components which are more attractive to human eyes. In this paper, we propose to set the sampling rate of each block based on texture details. We design texture contrast to measure the varying texture features and assign a high sampling rate to the block with a striking texture contrast. We remove the redundant CS samples of each block with a low-texture contrast. When reconstructing the image, the distribution of texture contrasts is used to weight the global reconstruction model. Experimental results show that the proposed method improves the visual quality of reconstructed image compared with the adaptive schemes based on block variance and edge features.

2. Adaptive Block CS of Image

The framework of adaptive block CS is shown in Figure 1. At CS encoder, the natural scene is first captured by CMOS sensors as a full-sampling image $\mathbf{x}$ with size of $I_r \times I_c$; that is, the total number of pixels N is $I_r \cdot I_c$. Then, divide image $\mathbf{x}$ into small blocks of $B \times B$ in size and let $\mathbf{x}_i$ represent the vectorized signal of the ith ($i = 1, 2, \ldots, n$, $n = N/B^2$) block through raster scanning. Next, the number M_i ($\ll B^2$) of CS samples for each block is set according to image features. We construct a random transformation matrix $\mathbf{\Phi}_{Bi}$ of $M_i \times B^2$ in size for each block. Finally, the CS-samples vector $\mathbf{y}_i$ of each block of M_i in length is computed by the following formulation:

$$\mathbf{y}_i = \mathbf{\Phi}_{Bi}\mathbf{x}_i, \tag{1}$$

in which the elements of $\mathbf{\Phi}_{Bi}$ obey Gaussian distribution. We define sampling rate S_i as follows:

$$S_i = \frac{M_i}{B^2}. \tag{2}$$

The CS-samples vectors of all blocks will be transmitted to the CS decoder. When receiving M_i CS samples of each block, we construct the minimum l_2-l_1 norm reconstruction model as follows:

$$\hat{\mathbf{x}}_i = \arg\min_{\mathbf{x}_i} \left\{ \left\| \mathbf{y}_i - \mathbf{\Phi}_{Bi}\mathbf{x}_i \right\|_2^2 + \lambda \left\| \mathbf{\Psi}_i \mathbf{x}_i \right\|_1 \right\}, \tag{3}$$

in which $\| \cdot \|_2$ and $\| \cdot \|_1$ are l_2 and l_1 norms, respectively, $\mathbf{\Psi}_i$ is the block transformation matrix, for example, DCT and wavelet matrices, and λ is a fixed regularization factor. Because the objective function of model (3) is convex, it can be solved by using Gradient Projection for Sparse Reconstruction (GPSR) algorithm [17] or Two-step Iterative Shrinkage Thresholding (TwIST) algorithm [18]. CS theory states that the signal can be recovered precisely by using model (3) if

$$M_i \geq c \cdot K_i \ln B^2, \tag{4}$$

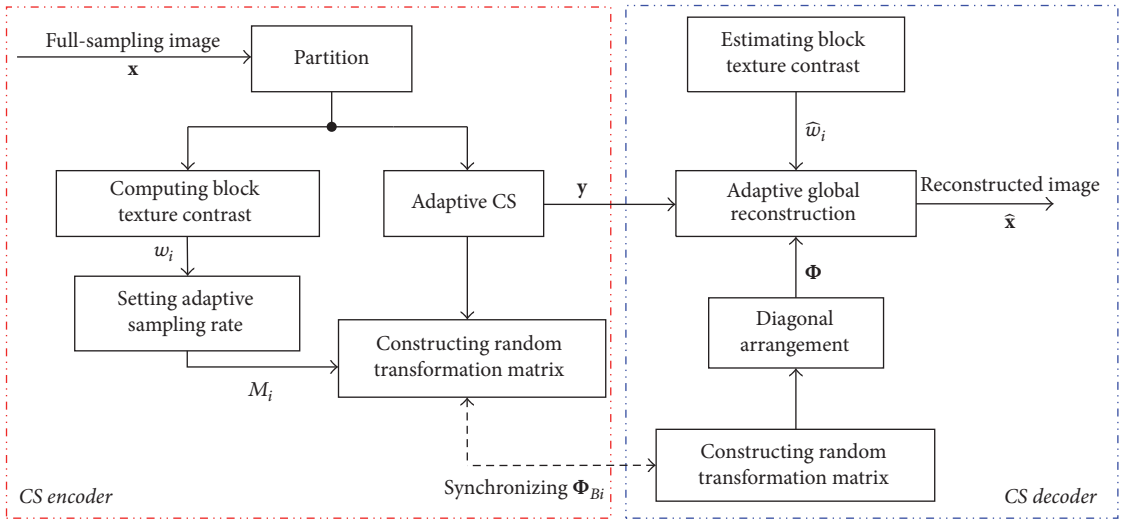


Figure 2: Flow of the proposed algorithm.

in which K_i is the sparse degree of the ith block and c is some constant [19]. Due to the nonstationary statistics of nature images, sparse degree of each block distributes nonuniformly. From (4), we can see that blocks with a large sparse degree cannot be accurately reconstructed once the sampling rate is too low; that is, the fixed number of block CS samples is not enough to capture all information on original image. Therefore, the sampling rate of each block should be set adaptively according to its own sparse degree. It is a straightforward method to acquire the block sparse degree that counts those significant transformation coefficients. However, this obviously violates the superiority of CS theory. Once the encoder performs full transformation, image CS has no advantage over the traditional coding. Therefore, it is impractical to directly get block sparse degree using full transformation. To avoid full transformation at encoder, some image features are exploited to indirectly reveal the block sparse degree, for example, block variance, the number of edge pixels. In this indirect way, we can get some improvement of rate-distortion performance; however, these features only reveal the varying of local pixel values, which improves the objective quality of reconstructed image but results in poor visual quality, which is shown especially by the occurrence of many blocking artifacts. In view of the above-mentioned, the proper feature is required to guide the adaptive sampling so as to improve the rate-distortion performance as well as guarantee a better visual quality.

3. Proposed Adaptive Sampling Scheme

Each block in nature image has different texture details. Rich-texture blocks are more attractive to human eyes due to the existence of more high-frequency components while low-texture blocks which have lots of low-frequency components tend to be neglected by human eyes. Therefore, uniform sampling will degrade the quality of reconstructed image. To solve the defect of uniform sampling, we propose to measure varying texture features of each block and then use them to guide the adaptive sampling and reconstruction. The flow of our method is presented in Figure 2. At CS encoder, we firstly generate the texture-feature map $\mathbf{v}_\alpha$ of full-sampling image $\mathbf{x}$. Then, we compute the block texture contrast w_i according to $\mathbf{v}_\alpha$. Afterward, the number M_i of block CS samples is determined adaptively by w_i. Finally, the partial transformation matrix $\mathbf{\Phi}_{Bi}$ is constructed to perform random sampling. At CS decoder, block texture contrast w_i is estimated again according to M_i. The estimated block texture contrast is used to weight the reconstruction model so as to improve visual quality of high-texture regions.

3.1. Computing Block Texture Contrast. The calculation of texture analysis should not be too much in order to guarantee low encoding complexity. To avoid excessive computations, we use the maximum gradient value in 8-connected region of each pixel to measure the texture variation of each pixel; that is,

$$v\left(x_{r,c}\right) = \max\left\{\left|x_{r,c} - x_{p,q}\right| \cdot \left|r-1 \le p \le r+1, c-1 \le q \le c+1\right\}\right., \tag{5}$$

in which $x_{r,c}$ is the luminance value at pixel position (r,c), $x_{p,q}$ is the luminance value at pixel position (p,q) in 8-connected region of $x_{r,c}$, and $|\cdot|$ is the operation to compute absolute value. The texture variation of each pixel in $\mathbf{x}$ can be computed by using (5) and used to construct the matrix $\mathbf{v}$ as follows:

$$\mathbf{v} = \begin{bmatrix} v\left(x_{1,1}\right) & \cdots & v\left(x_{1,I_c}\right) \\ \vdots & \ddots & \vdots \\ v\left(x_{1,I_r}\right) & \cdots & v\left(x_{I_r,I_c}\right) \end{bmatrix}. \tag{6}$$

(a) Original image (b) Block variance (c) Edge (d) Texture

Figure 3: Comparison of feature maps based on block variance, edge, and texture for 512 × 512 *Lenna*.

The matrix $\mathbf{v}$ is shrunk by hard-thresholding with threshold α to generate the texture-feature map $\mathbf{v}_\alpha$ as follows:

$$\mathbf{v}_\alpha\left[v\left(x_{r,c}\right)\right] = \begin{cases} v\left(x_{r,c}\right), & v\left(x_{r,c}\right) \geq \alpha \\ 0, & v\left(x_{r,c}\right) < \alpha, \end{cases} \tag{7}$$

in which the value α is set from 0 to 1. In the texture-feature map $\mathbf{v}_\alpha$, value 0 means no difference between current pixel and its neighbors, and value 1 means a big difference between current pixel and its neighbors. The energy of texture features in each block is computed as follows:

$$E_i = \frac{1}{B^2} \sum_{(r,c)\in\Lambda(\mathbf{x}_i)} v_\alpha\left[v\left(x_{r,c}\right)\right], \tag{8}$$

in which $\Lambda(\mathbf{x}_i)$ denotes the pixel position set of $\mathbf{x}_i$. We define the normalized texture-feature energy as the texture contrast, that is,

$$w_i = \frac{E_i}{\sum_{i=1}^{n} E_i}. \tag{9}$$

Figure 3 shows feature maps based on block variance, edge, and texture, among which the edge feature is extracted by using *Sobel* operator [20]. It can be seen that texture contrasts are highlighted in the region of hair and eye with rich texture details, and the edge features are also presented in the texture-feature map. However, maps of block variance and edge show fewer texture details, making features on block variance and edge not suitable to dominate the adaptive sampling which is meant to intensively capture the information on rich texture details. In view of the above analysis, the proposed texture contrast can guide CS sampling to capture the rich-texture block at a high sampling rate. As the core of traditional image coding, the fast DCT transformation bears computational complexity $O(N \log_2 N)$. For texture extraction, however, that is only $O(N)$, showing a low computational complexity at CS encoder to compute the texture contrast of each block.

3.2. Adaptive Sampling and Reconstruction. Due to nonstationary statistical characteristics of nature image, the block sampling rate varies with the texture contrast, introducing difficulties in controlling the bit rate. To handle that, we set a total sampling rate S for the whole image and then determine the total number M of CS samples as

$$M = N \cdot S, \tag{10}$$

in which N is the total pixel number. The number of CS samples for each block can be computed by using the block texture contrast w_i as follows:

$$M_i = \text{round}\left[w_i \cdot \left(M - nM_0\right) + M_0\right], \tag{11}$$

in which M_0 is the initial sampling number of each block and round$[\cdot]$ is the round operator. After determining the block sampling rate by (11), we assign some high-texture blocks to excessive CS samples, bringing a high-quality recovery of texture region. However, the blocks in non-texture region are assigned to fewer CS samples, leading to a worse reconstruction quality in non-texture region. The big difference of reconstruction quality makes texture region salient to human eyes, and thus degradation of visual quality happens. To solve that, we set the upper bound U of sampling number to be $0.9B^2$ for each block. Once the block sampling number exceeds the upper bound, its sampling number is limited to be U. The redundant CS samples are uniformly assigned to some blocks of which the sampling number is smaller than U. After reallocating the redundant CS samples, if the sampling numbers of some blocks exceed U again, we repeat the above steps until the sampling number of each block is smaller than U. According to the number M_i of block CS samples, we construct random transformation matrix $\mathbf{\Phi}_{Bi}$ and obtain block CS samples vector $\mathbf{y}_i$ by performing (1).

When the block CS samples $\mathbf{y}_i$ are received at CS decoder, (3) is used to reconstruct the image block by block. However, the minimum l_2-l_1 norm reconstruction model has different convergence performances for various block sparse degrees, giving rise to blocking artifacts of reconstructed image. To reduce blocking artifacts, we can perform the adaptive global reconstruction; that is, the image is recovered once by using

all block CS samples. First, all block CS samples are arranged in column as follows:

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_n \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi}_{B1} & & & \\ & \mathbf{\Phi}_{B2} & & \\ & & \ddots & \\ & & & \mathbf{\Phi}_{Bn} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{bmatrix}. \quad (12)$$

Suppose

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{\Phi}_{B1} & & & \\ & \mathbf{\Phi}_{B2} & & \\ & & \ddots & \\ & & & \mathbf{\Phi}_{Bn} \end{bmatrix}; \quad (13)$$

then we introduce the elementary matrix $\mathbf{I}$ to rearrange the column vectors block by block to a raster-scanning column vector of image as follows:

$$\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{bmatrix} = \mathbf{I} \cdot \mathbf{x}. \quad (14)$$

Combining (12), (13), and (14), we get

$$\mathbf{y} = \mathbf{\Phi} \cdot \mathbf{I} \cdot \mathbf{x} = \mathbf{\Theta} \cdot \mathbf{x}. \quad (15)$$

We construct a global reconstruction model as follows:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\{ \left\| \mathbf{y} - \mathbf{\Theta}\mathbf{x} \right\|_2^2 + \lambda \left\| \mathbf{\Psi}\mathbf{x} \right\|_1 \right\}, \quad (16)$$

in which $\mathbf{\Psi}$ is the transformation matrix of a whole image $\mathbf{x}$. With block sampling number M_i, which reveals the distribution of block texture contrast, we derive the estimator of block texture contrast from (11) as follows:

$$\widehat{w}_i = \frac{M_i - M_0}{M - nM_0}. \quad (17)$$

Using this estimator of block texture contrast, we weight the first term in (16) as follows:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\{ \sum_{i=1}^{n} \widehat{w}_i^2 \left\| \mathbf{y}_i - \mathbf{\Phi}_{Bi}\mathbf{x}_i \right\|_2^2 + \lambda \left\| \mathbf{\Psi}\mathbf{x} \right\|_1 \right\}. \quad (18)$$

By (18), we see that the larger $\widehat{w}_i$ prompts the random projection of block $\mathbf{x}_i$ to be closer to the CS samples vector $\mathbf{y}_i$. According to Johnson-Lindenstrauss (JL) theorem [21], the Euclidean distance between two blocks is similar to that between the corresponding CS-sample vectors [22]; that is, the weighting coefficients can enforce the rich-texture block to approach the original block and relax the requirement for the Euclidean distance between the low-texture block and its original. Therefore, this weighting constraint adaptively adjusts the reconstruction quality of each block according to the distribution of block texture contrasts. To simplify (18), we construct the diagonal matrix $\mathbf{W}$ as follows:

$$\mathbf{W} = \operatorname{diag}\left(\overbrace{\widehat{w}_1 \cdots \widehat{w}_1}^{M_1}, \cdots, \overbrace{\widehat{w}_i \cdots \widehat{w}_i}^{M_i}, \cdots, \overbrace{\widehat{w}_n \cdots \widehat{w}_n}^{M_n} \right), \quad (19)$$

in which diag$(\cdot)$ is an operator to generate diagonal matrix using the input vector. By using diagonal matrix $\mathbf{W}$, (18) is formed as

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\{ \left\| \mathbf{W} \cdot (\mathbf{y} - \mathbf{\Theta}\mathbf{x}) \right\|_2^2 + \lambda \left\| \mathbf{\Psi}\mathbf{x} \right\|_1 \right\}. \quad (20)$$

Suppose $\tilde{\mathbf{y}} = \mathbf{W}\mathbf{y}$ and $\mathbf{\Omega} = \mathbf{W}\mathbf{\Theta}$; we can get

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\{ \left\| \tilde{\mathbf{y}} - \mathbf{\Omega}\mathbf{x} \right\|_2^2 + \lambda \left\| \mathbf{\Psi}\mathbf{x} \right\|_1 \right\}. \quad (21)$$

We can see that the weighting reconstruction model is still the minimum l_2-l_1 norm model. Therefore, the traditional CS reconstruction algorithm can still be used to solve (21).

4. Experimental Results

Our method is evaluated on a number of grayscale images of 512 × 512 in size including *Lenna*, *Barbara*, *Peppers*, *Goldhill*, and *Mandrill*. These test images have different smooth, edge and texture details. For the adaptive sampling scheme, the parameters are set as follows: the initial sampling number M_0 of each block is set to be round$(0.3M/n)$, and the CS samples are quantized by 8-bit scalar quantization. For the adaptive reconstruction scheme, (21) is solved by using GPSR algorithm [17], and the transformation matrix $\mathbf{\Psi}$ uses Daubechies orthogonal wavelet of 4 in length. In all experiments, the block size B is set to be 8, and we set the total sampling rate S to be from 0.1 to 0.5. The Peak Signal to Noise Ratio (PSNR) between the reconstructed image and original image is used in the objective evaluation, but all PSNR values are averaged over 5 trials since the reconstruction quality varies with the randomness of random transformation matrix $\mathbf{\Phi}_{Bi}$. All experiments run under the following computer configuration: Intel(R) Core(TM) i3 @ 3.30 GHz CPU, 8 GB RAM, Microsoft Windows 7 32 bits, and MATLAB Version 7.6.0.324 (R2008a).

4.1. Select Threshold. In our adaptive sampling scheme, α is the only adjustable parameter. Figure 4 shows the impact of different α value on the PSNR value when sample rate S is 0.1, 0.3, and 0.5, respectively. It can be seen that each test image has higher PSNR values with S being 0.3 or 0.5 and α ranging from 0.1 to 0.3, which indicates that our method significantly improves the reconstruction quality at a moderate α value. However, when S is set to be 0.1, higher PSNR value appears with α being near 0.55, and the PSNR value lightly reduces with α increasing from 0.55, which indicates that α should be greater at a low sampling rate. α value is related to the richness of texture details. The greater α is, the more feature points gather in the rich-texture region, while conversely the feature points will spread to the edge and smooth regions. Therefore, when sampling rate S and threshold α are greater, the reconstruction quality of texture

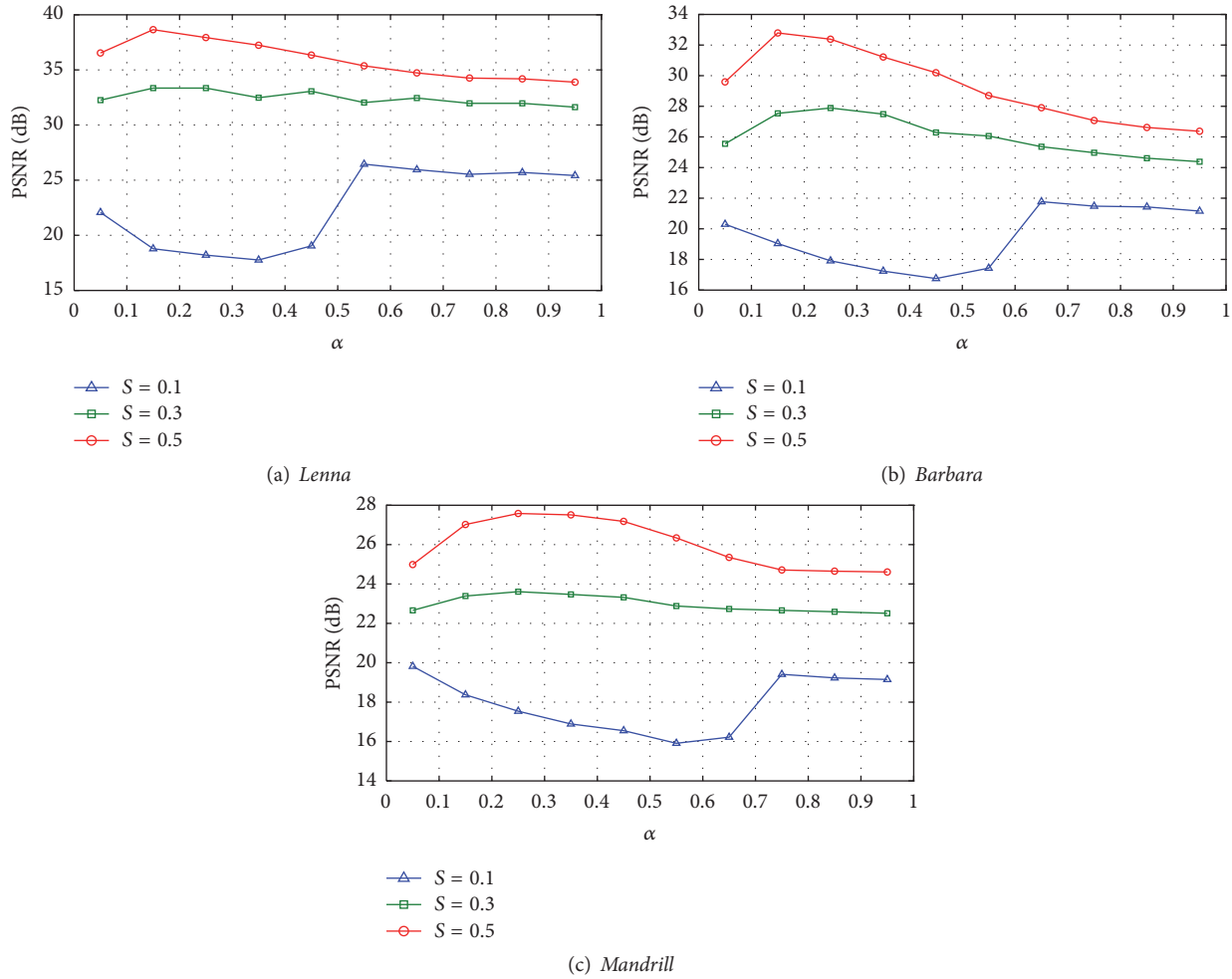


FIGURE 4: PSNR curves of the reconstructed images with the different threshold α when the sampling rate S is 0.1, 0.3, and 0.5, respectively.

region improves effectively, but otherwise for other regions, thus degrading the objective quality of a whole image. On the contrary, a small α could weaken the reconstruction quality of the texture region, which suppresses the improvement of reconstruction quality as well. When the sampling rate S is set to be small, limited by the number of CS samples, fewer CS samples can be assigned to the texture region once α value is low, so the reconstruction quality cannot be improved significantly. Apparently, the better objective quality requires a greater α. Given the above analysis, we set α at 0.15 in our adaptive scheme, in order to guarantee the robust reconstruction quality.

4.2. Performance Evaluation of Adaptive Sampling. Figure 5 shows the reconstructed *Lenna* images using different sampling schemes at CS encoder when S is set at 0.3. For the nonadaptive sampling scheme, some blurs occur in the reconstructed image, but edge and texture details cannot be well preserved. For adaptive sampling schemes, the block-variance based reconstructed image has obvious blocking artifacts. And it is the same with the edge based reconstructed one, though it is more visually pleasant than block-variance based reconstructed one. However, with blocking artifacts being suppressed, our scheme gets better visual quality. Besides, among the four schemes, our method obtains the highest PSNR value, 1.44 dB and 1.15 dB gains, respectively, compared with the block-variance and edge feature schemes.

4.3. Overall Performance Evaluation. To evaluate the rate-distortion performance of the proposed CS codec including adaptive sampling and reconstruction, we select sparse representation and quantization schemes as benchmarks. For the sparse representation scheme, scalar quantization is used to quantize CS samples; the benchmarks of evaluation are the Multi-Hypotheses Smoothed Projected Landweber (named as MH_SPL) algorithm proposed by [7] and the NESTA algorithm proposed by [8]. For the quantization scheme, the DPCM quantizer proposed by [12] is used as the benchmark of evaluation, and its corresponding reconstruction algorithm is the NESTA algorithm, which is named as DPCM + NESTA. The transformation matrix $\mathbf{\Psi}$ remains the same as the proposed algorithm. Figure 6 shows the average rate-distortion performance of different reconstruction algorithms. It can be seen that the proposed method improves the PSNR value as bit rate increases. When bit rate is higher than 1.3 bpp, the PSNR value of our method outperforms other algorithms, and the gap between them gradually increases. Table 1 lists the PSNR values of

(a) Original image

(b) Nonadaptive method (PSNR = 29.61 dB)

(c) Block-variance oriented method (PSNR = 32.12 dB)

(d) Edge oriented method (PSNR = 32.41 dB)

(e) Texture oriented method (PSNR = 33.56 dB)

Figure 5: Comparison of visual qualities of 512 × 512 *Lenna* sampled by different methods when the sampling rate S is 0.3.

Table 1: Comparison of PSNRs (dB) reconstructed images when using different algorithms.

	Lenna	*Barbara*	*Peppers*	*Goldhill*	*Mandrill*
		$S = 0.1$			
Sparse representation					
MH-SPL	25.12	21.05	24.41	24.07	19.23
NESTA	21.14	18.90	20.25	21.11	18.04
Quantization					
DPCM + NESTA	21.14	18.90	20.25	21.11	18.04
Proposed	21.04	19.13	20.90	21.14	18.07
		$S = 0.3$			
Sparse representation					
MH-SPL	29.89	24.91	30.08	28.19	21.54
NESTA	27.28	22.99	26.86	26.32	20.46
Quantization					
DPCM + NESTA	27.28	22.99	26.86	26.32	20.46
Proposed	33.32	27.92	32.69	30.51	23.67
		$S = 0.5$			
Sparse representation					
MH-SPL	32.93	27.74	33.21	30.72	23.50
NESTA	30.83	25.52	31.16	29.63	22.77
Quantization					
DPCM + NESTA	30.83	25.52	31.16	29.63	22.77
Proposed	37.79	32.49	36.40	34.61	27.79

Table 2: Comparison of execution time(s) to reconstruct an image when using different algorithms.

	Lenna	*Barbara*	*Peppers*	*Goldhill*	*Mandrill*
			$S = 0.1$		
Sparse representation					
MH-SPL	1.62	1.76	1.53	1.36	1.80
NESTA	198.47	174.76	197.28	115.32	159.45
Quantization					
DPCM + NESTA	224.06	174.07	184.70	191.80	156.42
Proposed	2.57	3.12	1.91	2.27	5.62
			$S = 0.3$		
Sparse representation					
MH-SPL	1.91	3.23	2.12	1.80	2.40
NESTA	134.34	170.11	130.25	130.48	117.25
Quantization					
DPCM + NESTA	152.26	116.98	126.01	125.42	114.77
Proposed	75.71	62.75	37.30	40.00	66.96
			$S = 0.5$		
Sparse representation					
MH-SPL	2.00	3.37	2.12	2.19	2.95
NESTA	120.89	102.80	117.19	110.17	196.12
Quantization					
DPCM + NESTA	120.00	100.81	110.05	107.47	98.36
Proposed	162.54	141.37	155.93	214.95	146.90

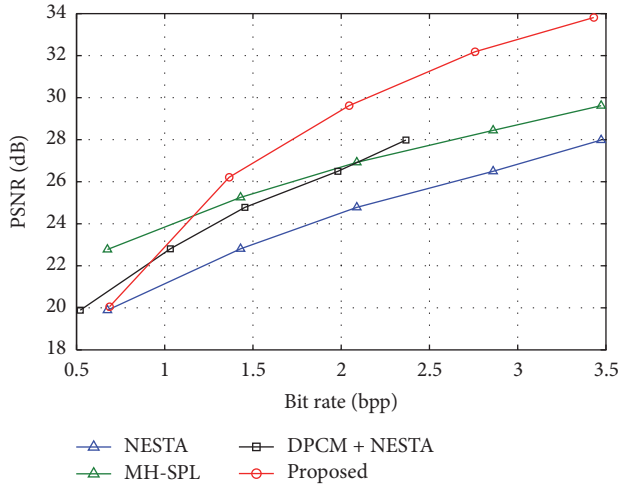
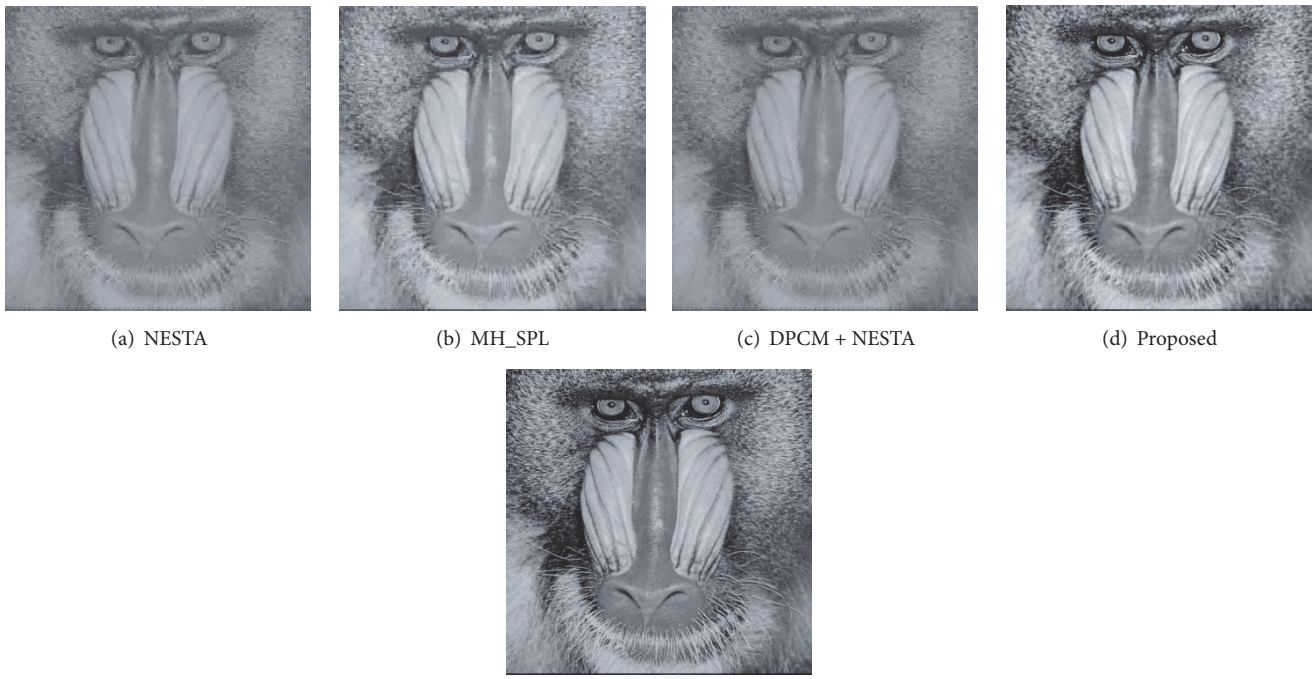


Figure 6: Comparison of average rate-distortion curves for the various reconstruction methods.

reconstructed images using different schemes at different sampling rates. It can be seen that the PSNR values of our method are lower than that of MH_SPL algorithm for each test image when the sampling rate S is 0.1, but the PSNR values of NESTA and DPCM + NESTA algorithms have little difference from that of our algorithm. However, when the sampling rate S is 0.3 or 0.5, our method achieves an obvious PSNR gain compared with other algorithms. Figure 7 shows the visual results of reconstructed *Mandrill* image by different methods, and it can be observed that the proposed method has better visual quality, especially that the texture details are better preserved when compared with other algorithms.

4.4. Computational Complexity. The proposed adaptive CS scheme involves extraction of texture feature, adaptive sampling, and global reconstruction. Suppose the total pixel number in test image is N, and the total number of CS samples is M. The extraction of texture feature takes $O(N)$ operations, and the computational complexity of adaptive sampling is $O(MN)$. We use GPRS algorithm to solve the global reconstruction model (21), in which transformation matrix $\mathbf{\Psi}$ is constructed by Daubechies orthogonal wavelet. According to the analysis of computational complexity in [17], the global reconstruction can be done $O(MN \log_2 N)$. Then we get the total computational complexity $O(N) + O(MN) + O(MN \log_2 N)$.

Table 2 lists the reconstruction times for different schemes at different sample rates. When the total sampling rate S is 0.1, the execution time of our method is close to that of MH_SPL for different test images, but much less than those of NESTA and DPCM + NESTA. As the total sampling rate S increases, the execution time of our method gradually increases as well. When the total sampling rate S is 0.5, our method takes 164.33 s on average to reconstruct an image. Compared with MH_SPL algorithm, our method requires more reconstruction time at a high sampling rate. Therefore, the improvement of PSNR value for our method requires a large amount of computation.



(a) NESTA (b) MH_SPL (c) DPCM + NESTA (d) Proposed

(e) Original

Figure 7: Comparison of visual qualities of *Mandrill* reconstructed by different methods when the sampling rate S is 0.3.

5. Conclusion

In this paper, we propose to adaptively sample and reconstruct images based on texture contrast. At the CS encoder, we first compute the texture contrast of each block, and then we set the sampling rate of each block adaptively according to the distribution of block texture contrasts. At CS decoder, the texture contrast of each block is used to weight the reconstruction model. Experimental results show that the proposed adaptive sampling and reconstruction algorithm can effectively improve the quality of reconstructed image. Our method has better rate-distortion performance than that of sparse representation and quantization schemes.

Image coding is the application background for our adaptive CS sampling scheme; thus full-sampling image is available at the encoder. However, since the encoder cannot use the full-sampling image for the application of compressive imaging, our method loses its efficacy. Therefore, our further study should be aimed at realizing the adaptive CS sampling in the analog domain for the sake of the application of our method in compressive imaging.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China, under Grants nos. 61501393 and 61601396, in part by the Key Scientific Research Project of Colleges and Universities in Henan Province of China, under Grant 16A520069, and in part by Youth Sustentation Fund of Xinyang Normal University, under Grant no. 2015-QN-043.

References

[1] S. Xie and Y. Wang, "Construction of tree network with limited delivery latency in homogeneous wireless sensor networks," *Wireless Personal Communications*, vol. 78, no. 1, pp. 231–246, 2014.

[2] P. Guo, J. Wang, X. H. Geng, C. S. Kim, and J.-U. Kim, "A variable threshold-value authentication architecture for wireless mesh networks," *Journal of Internet Technology*, vol. 15, no. 6, pp. 929–935, 2014.

[3] E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[4] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[5] C. Deng, W. Lin, B.-S. Lee, and C. T. Lau, "Robust image coding based upon compressive sensing," *IEEE Transactions on Multimedia*, vol. 14, no. 2, pp. 278–290, 2012.

[6] Z. Yu, X. Chen, S. Hoyos, B. M. Sadler, J. Gong, and C. Qian, "Mixed-signal parallel compressive spectrum sensing for cognitive radios," *International Journal of Digital Multimedia Broadcasting*, vol. 2010, Article ID 730509, 10 pages, 2010.

[7] C. Chen, E. W. Tramel, and J. E. Fowler, "Compressed-sensing recovery of images and video using multihypothesis predictions," in *Proceedings of the 45th Asilomar Conference on Signals,*

Systems and Computers (ASILOMAR '11), pp. 1193–1198, Pracific Grove, Calif, USA, November 2011.

[8] S. Becker, J. Bobin, and E. J. Candes, "NESTA: a fast and accurate first-order method for sparse recovery," *SIAM Journal on Imaging Sciences*, vol. 4, no. 1, pp. 1–39, 2011.

[9] J. Zhang, D. Zhao, and W. Gao, "Group-based sparse representation for image restoration," *IEEE Transactions on Image Processing*, vol. 23, no. 8, pp. 3336–3351, 2014.

[10] J. Yang, X. Liao, X. Yuan et al., "Compressive sensing by learning a Gaussian mixture model from measurements," *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 106–119, 2015.

[11] L. Wang, X. Wu, and G. Shi, "Binned progressive quantization for compressive sensing," *IEEE Transactions on Image Processing*, vol. 21, no. 6, pp. 2980–2990, 2012.

[12] S. Mun and J. E. Fowler, "DPCM for quantized block-based compressed sensing of images," in *Proceedings of the 20th European Signal Processing Conference (EUSIPCO '12)*, pp. 1424–1428, Bucharest, Romania, August 2012.

[13] J. Zhang, D. Zhao, and F. Jiang, "Spatially directional predictive coding for block-based compressive sensing of natural images," in *Proceedings of the 20th IEEE International Conference on Image Processin (ICIP '13)*, pp. 1021–1025, Melbourne, Australia, September 2013.

[14] X. Gao, J. Zhang, W. Che, X. Fan, and D. Zhao, "Block-based compressive sensing coding of natural images by local structural measurement matrix," in *Proceedings of the Data Compression Conference (DCC '15)*, pp. 133–142, Snowbird, Utah, USA, April 2015.

[15] J. Zhang, Q. Xiang, Y. Yin, C. Chen, and X. Luo, "Adaptive compressed sensing for wireless image sensor networks," *Multimedia Tools and Applications*, vol. 76, no. 3, pp. 4227–4242, 2017.

[16] T. N. Canh, K. Q. Dinh, and B. Jeon, "Edge-preserving nonlocal weighting scheme for total variation based compressive sensing recovery," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '14)*, pp. 1–5, Chengdu, China, July 2014.

[17] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 586–597, 2007.

[18] J. M. Bioucas-Dias and M. A. T. Figueiredo, "A new TwIST: two-step iterative shrinkage/thresholding algorithms for image restoration," *IEEE Transactions on Image Processing*, vol. 16, no. 12, pp. 2992–3004, 2007.

[19] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling: a sensing/sampling paradigm that goes against the common knowledge in data acquisition," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[20] W. Gao, X. Zhang, L. Yang et al., "An improved Sobel edge detection," in *Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology*, pp. 67–71, Chengdu, China, 2010.

[21] S. Dasgupta and A. Gupta, "An elementary proof of a theorem of Johnson and Lindenstrauss," *Random Structures & Algorithms*, vol. 22, no. 1, pp. 60–65, 2003.

[22] K. Zhang, L. Zhang, and M.-H. Yang, "Fast compressive tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 10, pp. 2002–2015, 2014.