2016 Data Compression Conference

Compressive-Sensed Image Coding via Stripe-based DPCM

Chen Zhao†, Jian Zhang†, Siwei Ma†‡, and Wen Gao†‡

†Institute of Digital Media & Cooperative Medianet Innovation Center
Peking University, Beijing, 100871, P.R. China
‡Peking University Shenzhen Graduate School, Shenzhen, 518055, P. R. China
{zhaochen,jian.zhang,swma,wgao}@pku.edu.cn

Abstract

These years have seen the advances of compressive sensing (CS), but efficient coding of sensed measurements is still an issue. In this paper, we propose an image coding system based on the compressive sensing paradigm via stripe-based differential pulse-code modulation (DPCM). In the system, we sample and encode an image in a unit of multiple rows, which we call a *stripe*. Through extensive experiments, we observe that the correlation between measurements of adjacent stripes are much higher than that of neighboring blocks. Based on this, we combine the stripe-based CS acquisition with the DPCM framework and design a mechanism that predicts a stripe of measurements from its preceding stripe of measurements. The produced measurement residuals are then quantized and entropy-encoded into binary coding bits, which are tremendously reduced compared to the traditional block-based framework. Furthermore, we provide an image CS reconstruction algorithm corresponding to the stripe-based acquisition. Experiments verify that the reconstruction quality is no worse or even better than the block-based case when much lower bitrate is consumed. In a rate-distortion point of view, the proposed system also outperforms the methods using block-based sampling and achieves the state-of-the-art performance for compressive-sensed image coding.

1 Introduction

With the development of the compressive sensing (CS) theory and application, more and more interest is devoted into this area. Compressive sensing features far fewer measurements for a signal than those required by the Shannon-Nyquist theory; by projecting the original signal into a lower-dimentional subspace chosen at random, it can be accurately recovered under certain conditions [1]. Thus, the coding system based on compressive sensing seems to accomplish compression while sampling and does not require the complex compression procedures in conventional coding paradigm [2]. Strictly speaking, however, the measurements still contain a lot of redundant data in an aspect of the information theory. Therefore, coding techniques such as quantization and entropy coding are necessary so that the CS system can achieve equivalent or higher rate-distortion performance than conventional coding methods [3].

There have been a few works in the literation for compressive-sensed image coding. A straightforward method is to do a scalar quantization (SQ) [3] for each of the sampled measurements in an image. Although this reduces coding bits by dropping some insignificant information in the measurements, it is proved to be inefficient

1068-0314/16 $31.00 © 2016 IEEE
DOI 10.1109/DCC.2016.103

in a rate-distortion perspective compared to conventional coding methods. Thus, by collaboratively considering the coding bits and the reconstruction performance, some methods attempting to optimize the quantization or reconstruction process were proposed in the literature (e.g., [4], [5]).

The introduction of the block-based compressive sensing, which tackles the issues of memory and computation burden for real implementation, brings new possibilities for measurement coding. In this framework, an image is divided into non-overlapped blocks, which are sensed independently using a comparatively small matrix. Based on this, S. Mun and J. E. Fowler propose to use differential pulse code modulation (DPCM) of the CS measurements in conjunction with the uniform scalar quantization [6]. Instead of directly quantizing each measurement, they first produce a prediction for the current block using measurements of the preceding neighboring block. Then it is subtracted from the current block in the measurement domain to generate residuals, which are scalar-quantized uniformly. This method outperforms scalar quantization and rivals JPEG in some cases.

Although predictive coding using neighboring blocks achieves better performance than scalar quantization alone, the correlation between neighboring blocks is limited if we take a closer look. A block is in a square shape, both the row and the column having the same number of pixels. Taking a pair of horizontally neighboring blocks for example, the left-most pixel of the left block and the right-most pixel of the right block are $2S - 1$ pixels away from each other, if the block size is $S \times S$. It is very probable that they are completely different when S is not small enough. Consequently, DPCM cannot play an efficient role with a dissimilar prediction. In order to address this issue, rather than dividing an image into blocks, we design a novel sampling unit called a stripe, which consists of multiple rows of pixels. A stripe is much flatter than a block, thus the pixel distance between neighboring stripes is much smaller and the correlation is much higher accordingly. To be specific, for a stripe size $2 \times N$, the farthest distance between pixels of neighboring stripes is only 3. Besides the good predictive property of the stripe design, many applications favoring linear sensing would benefit from it, such as flatbed scanners and airborne spaceborne images [7].

Based on our stripe design, we propose a CS image coding system in this paper, which is organized as follows. We first give a background introduction of the compressive sensing paradigm in Section 2. In Section 3, the proposed method for compressive-sensed image coding is presented in detail. Section 4 provides simulation results to demonstrate the performance of the method and Section 5 concludes this paper.

2 Background

The compressive sensing (CS) theory states that a signal can be accurately recovered under certain conditions after being projected into a much lower dimension[1]. Concretely, let us consider a signal of a finite dimension $\mathbf{x} \in \mathbb{R}^N$. The acquisition process of CS is expressed as

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

where $\mathbf{\Phi}$ is an $M \times N$ random projection matrix and $\mathbf{y} \in \mathbb{R}^M$ represents the acquired linear measurements. M/N is called the sampling rate or subrate, which is typically very small. Since the number of observations are far fewer than the unknowns, the inverse problem is highly ill-posed and impossible to solve in general. But according to the CS theory, if the original signal is sufficiently sparse in a transform domain, its exact recovery is possible.

We represent the signal $\mathbf{x}$ in an appropriate $N \times N$ basis $\mathbf{\Psi}$, i.e., $\mathbf{x} = \mathbf{\Psi}\boldsymbol{\alpha}$. If at most $K \ll N$ entries of the coefficients $\boldsymbol{\alpha}$ are nonzero, we say $\mathbf{x}$ is K-sparse in the domain $\mathbf{\Psi}$. Although many natural signals are not strictly sparse, they can be approximated as such; we call them compressible signals. According to the CS theory, sparse and compressible signals can be reconstructed by solving the following minimization problem:

$$\min_{\mathbf{x}} \|\phi(\mathbf{x})\|_0, \quad \text{s.t. } \mathbf{y} = \mathbf{\Phi}\mathbf{x}, \tag{2}$$

where $\phi(\mathbf{x})$ denotes a prior property of the signal, usually a sparsity representation. By converting the above unconstrained problem to a constrained problem by introducing a penalty parameter λ, we get

$$\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{y} - \mathbf{\Phi}\mathbf{x}\|_2^2 + \lambda \|\phi(\mathbf{x})\|_0 . \tag{3}$$

This is the general paradigm of CS-based sampling and reconstruction processes. For real application of CS-based coding, the measurements should be encoded before being put into the transmission channel. In the next section, we provide our proposed coding system and the detailed techniques.

3 Proposed Compressive-Sensed Image Coding Method

In this section, we first introduce the architecture design of our proposed image coding system in Part 3.1. Then we detail the stripe-based image acquisition mechanism in Part 3.2 and discuss the predictive coding with the stripe acquisition mechanism in Part 3.3. Finally in Part 3.4, we give our reconstruction algorithm.

3.1 Architecture of the Proposed Image Coding System

The proposed coding system combines the idea of stripe-based sampling and differential pulse code modulation (DPCM). Fig. 1 shows its overall architecture.

In the encoder side, an input image is first divided into non-overlapped stripes, each consisting of multiple rows of image pixels. Each stripe of pixels is projected into the measurement domain by being applied a random matrix independently. Then all stripes of measurements are put into a DPCM encoder, in which the measurements of one stripe are predicted from those of its previous stripe and the residuals are then scalar-quantized and entropy-encoded to generate a binary bitstream. The binary bitstream is transmitted over a channel and reaches the decoder side.

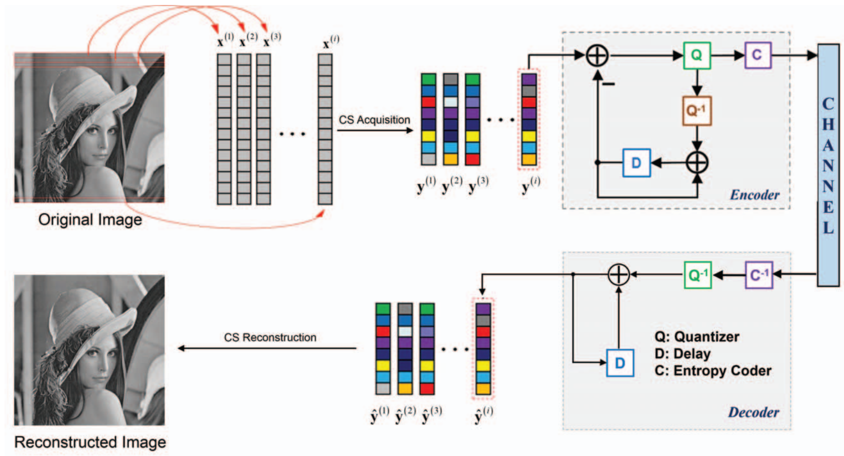

Figure 1: The architecture of the proposed CS image coding system. Q is uniform scalar quantization; Q^{-1} is inverse uniform scalar quantization; C is an entropy encoder; C^{-1} is an entropy decoder; D refers to a delay buffer containing measurements of the previous stripe.

In the decoder side, the bitstream is first interpreted by an entropy decoder and is de-quantized using the same quantization parameters as the encoder. The measurement residuals are hereby recovered and added to the already-recovered measurements of the previous stripe to produce the measurements of the current stripe. With the recovered measurements of all stripes, we apply a CS reconstruction algorithm and finally obtain the reconstructed image.

3.2 Stripe-based Image Acquisition

Suppose that the scene we wish to acquire (the original image) is represented by the matrix $\mathbf{X} \in \mathbb{R}^{M \times N}$. If we make a stripe consist of s rows of image pixels, then the image could be divided into M/s non-overlapped stripes. All pixels in a stripe are sampled simultaneously by being projected into a lower dimension via a random matrix. The acquisition process for the i-th stripe in the image, where $1 \leq i \leq M/s$, is formulated mathematically as follows

$$\mathbf{y}^{(i)} = \mathbf{\Phi}\mathbf{x}^{(i)}, \tag{4}$$

where the vector $\mathbf{x}^{(i)} \in \mathbb{R}^{sN \times 1}$ denotes the i-th stripe of pixels, which are rearranged in a horizontal scan order into a column vector. The matrix $\mathbf{\Phi} \in \mathbb{R}^{m \times sN}$ is the projection matrix, which is composed of Gaussian random numbers as its elements. The vector $\mathbf{y}^{(i)} \in \mathbb{R}^{m \times 1}$ represents the obtained measurements for the i-th stripe. For the whole image, this sampling process is performed progressively from top to down, which is summarized in Table 1.

Table 1: The proposed stripe-based CS acquisition process for an input image

Algorithm: Stripe-based image acquisition
1: **Input:** the image $\mathbf{x}$, which is a column vector for $\mathbf{X}$; the projection matrix $\mathbf{\Phi}$
2: Divide the input image into non-overlapped stripes by $\mathbf{x}^{(i)} = \mathbf{E}^{(i)}\mathbf{x}$, where $\mathbf{E}^{(i)}$ is a matrix operator that extracts the i-th stripe $\mathbf{x}^{(i)}$ from $\mathbf{x}$
3: **For** $i = 1$ to M/s **Do**
4: Calculate Eq. (4) to get the measurements for the i-th stripe
5: **End For**
6: **Output:** the measurements of all the stripes $\mathbf{Y} = \left[\mathbf{y}^{(1)}\mathbf{y}^{(2)}\ldots\mathbf{y}^{(M/s)}\right]$

Figure 2: Demonstration of the correlation between blocks and stripes. Each block in the left image has the same number of pixels as each stripe in the right image. In the left image, the green block is adjacent to the red block, but they are quite different in content and texture. In contrast, the two neighboring stripes in the right image – the green one and the red one are quite similar in pixel values.

3.3 Stripe-based Predictive Coding

Our CS image coding system takes advantage of DPCM, which works well when neighboring signal segments exhibit high correlation. Consecutive stripes naturally have higher similarity than consecutive blocks in the pixel domain, which is illustrated in Fig. 2.

The high correlation between neighboring stripes of image pixels are reserved in the random measurement domain by $\mathbf{y}^{(i)} = \mathbf{\Phi}\mathbf{x}^{(i)}$. To further verify this, we also provide a quantitative comparison between the measurement correlations of neighboring blocks and of neighboring stripes. As in [6], we compute the correlation coefficient to evaluate the correlation. Assuming the measurement vector for a block or a stripe is denoted as $\mathbf{y}^{(i)}$, then its correlation coefficient with another measurement vector $\mathbf{y}^{(j)}$ is formulated as

$$\rho\left(\mathbf{y}^{(i)}, \mathbf{y}^{(j)}\right) = \frac{\mathbf{y}^{(i)T}\mathbf{y}^{(j)}}{\|\mathbf{y}^{(i)}\| \cdot \|\mathbf{y}^{(j)}\|} \tag{5}$$

According to Eq. (5), we calculate the average correlation coefficients (ACC) over all pairs of neighboring blocks and all pairs of neighboring stripes in an image respectively. They are written as

Table 2: Average correlation coefficient for neighboring measurement blocks/stripes

Pixel number	ACC	Clown	Peppers	Lena	Average
0.5×10^3	ACC_B	0.8063	0.8945	0.9256	0.8755
	ACC_S	0.9946	0.9971	0.9982	**0.9966**
1×10^3	ACC_B	0.7955	0.9090	0.9424	0.8823
	ACC_S	0.9841	0.9930	0.9946	**0.9905**
2×10^3	ACC_B	0.7206	0.8464	0.8708	0.8126
	ACC_S	0.9611	0.9854	0.9882	**0.9783**
3×10^3	ACC_B	0.7050	0.8215	0.8486	0.7917
	ACC_S	0.9364	0.9724	0.9777	**0.9621**
4×10^3	ACC_B	0.7319	0.8800	0.9163	0.8427
	ACC_S	0.9236	0.9699	0.9782	**0.9573**
5×10^3	ACC_B	0.6356	0.7417	0.7655	0.7143
	ACC_S	0.8961	0.9515	0.9626	**0.9367**

$$ACC_B = \frac{1}{N_{block}} \sum_{1 \le i \le N_{block}-1} \rho\left(\mathbf{y}_{block}^{(i)}, \mathbf{y}_{block}^{(i+1)}\right), \tag{6}$$

and

$$ACC_S = \frac{1}{N_{stripe}} \sum_{1 \le i \le N_{stripe}-1} \rho\left(\mathbf{y}_{stripe}^{(i)}, \mathbf{y}_{stripe}^{(i+1)}\right), \tag{7}$$

respectively. N_{block} denotes the total number of blocks in an image; $\mathbf{y}_{block}^{(i)}$ and $\mathbf{y}_{block}^{(i+1)}$ are measurement vectors of two neighboring blocks. N_{stripe} denotes the total number of stripes in an image; $\mathbf{y}_{stripe}^{(i)}$ and $\mathbf{y}_{stripe}^{(i+1)}$ are measurement vectors of two neighboring stripes.

We test the values of ACC_B and ACC_S on a variety of images. Table 2 demonstrates the results of three images (the subrate is 0.5). In the stripe case, we test six different stripe sizes $1 \times 512, 2 \times 512, 4 \times 512, 6 \times 512, 8 \times 512$ and 10×512; and in the block case, to make the pixel number in a block as close as possible to the pixel number in a stripe, we use the corresponding six different block sizes $23 \times 23, 32 \times 32, 45 \times 45, 55 \times 55, 64 \times 64$ and 72×72. *Pixel number* in Table 2 refers the round pixel number in one stripe or block. We can see that ACC_S is obviously higher than ACC_B for all pixel numbers. This indicates that prediction using stripe-based CS acquisition is more accurate than using block-based CS acquisition, and thus smaller residuals would be produced. As a result, a lower bitrate is generated from the DPCM encoder.

3.4 CS reconstruction algorithm of the proposed coding system

In the decoder side, the measurements of all stripes are recovered after the received bitstream passes through a DPCM decoder. We need to reconstruct the whole image $\mathbf{X}$ using the decoded measurements $\hat{\mathbf{Y}} = \left[\hat{\mathbf{y}}^{(1)}\hat{\mathbf{y}}^{(2)} \cdots \hat{\mathbf{y}}^{(M/s)}\right]$. For this purpose, we design an optimization algorithm for the following reconstruction problem

Table 3: The proposed stripe-based CS reconstruction algorithm for an input image

Algorithm: Stripe-based image reconstruction
1: **Input:** the recovered measurements $\hat{\mathbf{Y}}$, the projection matrix $\mathbf{\Phi}$
2: Initialize the reconstructed image vector $\hat{\mathbf{x}}$
3: **For** $k = 1$ to $Max\_iteration$ **Do**
4: $\mathbf{r} \leftarrow \hat{\mathbf{x}} - \eta \sum_{1 \le i \le M/s} \mathbf{R}^{(i)T} \left(\mathbf{R}^{(i)} \hat{\mathbf{x}} - \hat{\mathbf{y}}^{(i)} \right)$, where $\mathbf{R}^{(i)} = \mathbf{\Phi} \mathbf{E}^{(i)}$
5: $\hat{\mathbf{x}} \leftarrow \arg\min_{\mathbf{x}} \frac{1}{2} \left\| \mathbf{x} - \mathbf{r} \right\|_2^2 + \lambda \left\| \phi(\mathbf{x}) \right\|_0$
6: **End For**
7: Rearrange $\hat{\mathbf{x}}$ to obtain the image $\hat{\mathbf{X}}$
8: **Output:** the reconstructed image $\hat{\mathbf{X}}$

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2} \sum_{1 \le i \le M/s} \min_{\mathbf{x}} \left\| \hat{\mathbf{y}}^{(i)} - \mathbf{\Phi} \mathbf{E}^{(i)} \mathbf{x} \right\|_2^2 + \lambda \left\| \phi(\mathbf{x}) \right\|_0, \tag{8}$$

where $\mathbf{x} \in \mathbb{R}^{MN \times 1}$ is a column vector containing all the pixel values in the original image $\mathbf{X}$ and $\hat{\mathbf{x}}$ is its reconstructed value. $\mathbf{E}^{(i)}$ is a matrix operator that extracts the i-th stripe $\mathbf{x}^{(i)}$ from $\mathbf{x}$, so we have $\mathbf{x}^{(i)} = \mathbf{E}^{(i)} \mathbf{x}$. Using the idea of the iterative shrinkage/thresholding algorithm (ISTA) [8], we solve Eq. (8) by alternating two iterative steps, as shown in Table 3.

In Table 3, the specific solution for the equation in Line 5 is dependent on the form of $\phi(\mathbf{x})$. For example, if $\phi(\mathbf{x})$ represents the wavelet sparsity of an image, i.e., $\phi(\mathbf{x}) = \mathbf{\Psi} \mathbf{x}$, where $\mathbf{\Psi}$ is the wavelet transform matrix, then Line 5 is replaced by $\hat{\mathbf{x}} \leftarrow hard\left(\mathbf{r}, \sqrt{2\lambda} \right)$, where $hard$ refers to the hard-thresholding algorithm [9]. Other efficient image reconstruction models (e.g. [10], [11]) can also be incorporated into this framework.

In Part 3.3, we showed the bit-saving property of the stripe-based strategy. One may wonder whether it would bring down the reconstruction quality without as many coding bits. To resolve this concern, we compare the reconstruction performance of the proposed stripe-based system with the performance of the block-based system. In order to eliminate the influence of quantization to the final reconstruction results, we do this comparison in a lossless setting, i.e., disabling scalar quantization in both systems. We apply the same sparsity model for both systems, the DDWT sparsity model [12] to be specific. The settings of the block sizes and the stripe sizes are the same as those in Table 2. Five different subrates are adopted: 0.2, 0.25, 0.3, 0.35, and 0.4. Fig. 3 demonstrates the results for the test image 'Lena' of the size 512×512. We can see that the reconstruction performance of the stripe-based system is no worse or even better than the block-based system except for the upper-left test case. This exceptional inferior performance is because there is only one row of pixels in a stripe, which is not stable for reconstruction. This is also the reason that we make our processing unit a stripe, which is composed of multiple rows, rather than a single row.

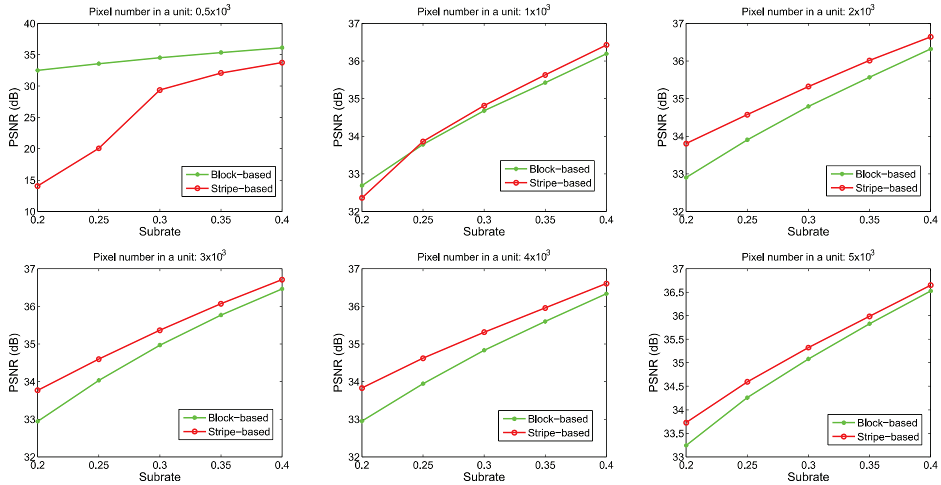

Figure 3: Reconstruction comparison of the block-based and the stripe-based CS systems for 'Lena'.

4 Simulation Results

In this section, we provide simulation results to verify the rate-distortion (RD) performance of the proposed stripe-based CS image coding method. The RD performance is in terms of peak signal-to-noise ratio (PSNR) in dB versus bitrate in bits per pixel (bpp). Following [6], we estimate the bitrate using the entropy of the quantizer indices, which could be actually produced by a real entropy coder. Three images of the size 512×512 are tested: 'Lena', 'Pepppers', and 'Clown'.

We first consider which stripe size is optimal for our system to reach its best performance. If the size is too large, then the correlation between neighboring stripes would not be as high; if the size is too small (e.g., 1 row of pixels), it is not stable for reconstruction. We test the RD performances of five different stripe sizes and demonstrate the results in Fig. 4. We can see that when the size is 2×512, the proposed method produces the highest RD quality.

Then we compare its RD performance with the methods of scalar quantization (SQ) and the block-based DPCM [6]. For all the methods, the DDWT sparsity model [12] is utilized for reconstruction and the same setup for the combination of quantization step-size and subrate is utilized as in [6]. There is also a tradeoff in selecting the block size for the two comparative methods. Ref [13] suggests a block size of 32×32, and Ref [6] utilizes a block size of 16×16 to demonstrate its performance. We compare the RD performances of the block sizes $16 \times 16, 32 \times 32, 48 \times 48, 64 \times 64, 70 \times 70$, and find 16×16 is the best for both methods. Fig. 5 demonstrates the RD performance comparison of the proposed method with the stripe size 2×512 and the methods of SQ and the block-size DPCM with the block size 16×16. We can see that the the proposed method outperforms the other methods by obvious gains and achieves the best RD performance.

In Figure 6, we provide the visual comparison of the proposed method and the

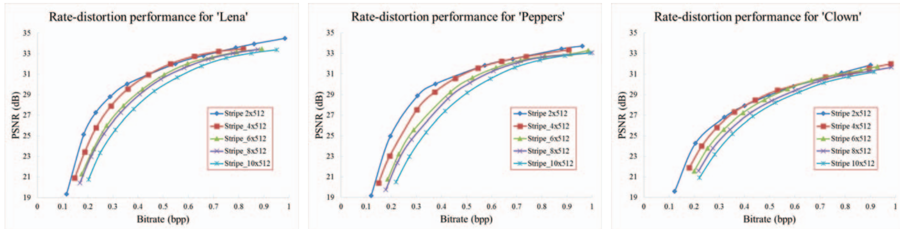

Figure 4: Rate-distortion performance of the proposed CS system with different stripe sizes: $2 \times 512, 4 \times 512, 6 \times 512, 8 \times 512$ and 10×512. The DDWT sparsity model [12] is utilized for reconstruction. The same setup for the combination of quantization step-size and subrate is utilized as in [6].

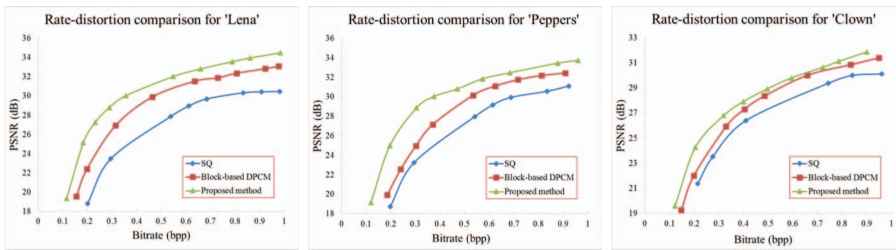

Figure 5: Rate-distortion performance comparison of the proposed CS system and the methods of SQ and block-based DPCM.

block-based DPCM at a bitrate of 0.2 bpp. We can see that the reconstructed images using the proposed method have less noises and are much more visually pleasant.

5 Conclusion

In this paper, we propose a novel compressive-sensed image coding method. We bring up the concept of a *stripe*, and by incorporating it into the DPCM framework, we design a CS image acquisition-reconstruction system. We verify the effectiveness of the stripe-based system in the aspects of measurement correlation, reconstruction quality and the rate-distortion performance. Experiments demonstrate that the pro-

Figure 6: Visual comparison at a low bitrate of 0.2 bpp for the two test images: 'Peppers' and 'Lena'. The first and the third images show the results of the block-based DPCM with the block size 16×16, and the second and the third images show the results of the proposed method with the stripe size 2×512.

posed method brings significant improvement compared to the block-based methods in both rate-distortion objective performance and the visual subjective results.

This new design of a CS system breaks the square-shape restriction of a sampling unit and implies more flexible unit-division. For future work, other shapes of a unit could be considered, and besides, more efficient sparsity models corresponding to this paradigm can be designed to further improve the reconstruction quality.

6 Acknowledgement

This work was supported in part by the Major State Basic Research Development Program of China (2015CB351800), in part by the National Science Foundation (No. 61322106 and No. 61572047), and Shenzhen Peacock Plan, which are gratefully acknowledged.

References

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

[2] Y. C. Eldar and G. Kutyniok, *Compressed sensing: theory and applications.* Cambridge University Press, 2012.

[3] V. K. Goyal, A. K. Fletcher, and S. Rangan, "Compressive sampling and lossy compression," *Signal Processing Magazine, IEEE*, vol. 25, no. 2, pp. 48–56, 2008.

[4] J. Z. Sun and V. K. Goyal, "Optimal quantization of random measurements in compressed sensing," in *Information Theory, 2009. ISIT 2009. IEEE International Symposium on.* IEEE, 2009, pp. 6–10.

[5] P. T. Boufounos and R. G. Baraniuk, "1-bit compressive sensing," in *Information Sciences and Systems, 2008. CISS 2008. 42nd Annual Conference on.* IEEE, 2008, pp. 16–21.

[6] S. Mun and J. E. Fowler, "DPCM for quantized block-based compressed sensing of images," in *Proc. of the Euro. Signal Process. Conf*, 2012, pp. 1424–1428.

[7] G. Coluccia and E. Magli, "A novel progressive image scanning and reconstruction scheme based on compressed sensing and linear prediction," in *Multimedia and Expo (ICME), 2012 IEEE International Conference on.* IEEE, 2012, pp. 866–871.

[8] M. V. Afonso, J. M. Bioucas-Dias, and M. A. Figueiredo, "Fast image recovery using variable splitting and constrained optimization," *Image Processing, IEEE Transactions on*, vol. 19, no. 9, pp. 2345–2356, 2010.

[9] J. Zhang, D. Zhao, C. Zhao, R. Xiong, S. Ma, and W. Gao, "Image compressive sensing recovery via collaborative sparsity," *Emerging and Selected Topics in Circuits and Systems, IEEE Journal on*, vol. 2, no. 3, pp. 380–391, 2012.

[10] 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.

[11] J. Zhang, D. Zhao, R. Xiong, S. Ma, and W. Gao, "Image restoration using joint statistical modeling in a space-transform domain," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 6, pp. 915–928, 2014.

[12] S. Mun and J. E. Fowler, "Block compressed sensing of images using directional transforms," in *Image Processing (ICIP), 2009 16th IEEE International Conference on.* IEEE, 2009, pp. 3021–3024.

[13] L. Gan, "Block compressed sensing of natural images," in *Digital Signal Processing, 2007 15th International Conference on.* IEEE, 2007, pp. 403–406.