Joint Decoding of Independently Encoded Compressive Multi-view Video Streams

Nan Cen[1], Zhangyu Guan[1,2], Tommaso Melodia[1]
[1]Department of Electrical Engineering
The State University of New York (SUNY) at Buffalo, Buffalo, NY, 14260
[2]School of Information Science and Engineering
Shandong University, Jinan, China, 250100
Email:{nancen, zguan2, tmelodia}@buffalo.edu

***Abstract*—We design a video coding and decoding framework for multi-view video systems based on compressed sensing imaging principles. Specifically, we focus on joint decoding of independently encoded compressively-sampled multi-view video streams. We first propose a novel distributed coding/decoding architecture designed to leverage inter-view correlation through joint decoding of the received compressively-sampled frames. At the encoder side, we select one view (referred to as K-view) as a reference for the other views (referred to as CS-views). The video frames of the CS-view are encoded and transmitted at a lower measurement rate than those of the selected K-view. At the decoder side, we generate side information to decode the CS-views as follows. First, each K-view frame is down-sampled and reconstructed, and then compared with the initially reconstructed CS-view frame to obtain an estimate of the inter-view motion vector. The original CS-view measurements are then fused with the generated side image to reconstruct the CS-view frame through a newly designed algorithm that operates in the measurement domain.**

We also propose a blind video quality estimation method that can be used within the proposed framework to design channel-adaptive rate control algorithms for quality-assured multi-view video streaming. We extensively evaluate the proposed scheme using real multi-view video traces. Results indicate that up to 1.6 dB improvement in terms of PSNR can be achieved by the proposed scheme compared with traditional independent decoding of CS frames.

I. Introduction

Compressive sampling (CS) can be used to reconstruct image signals from a "small" number of (random or deterministic) linear combinations, referred to as measurements or samples, of the original image pixels without collecting the entire frame [1], [2]. CS-based imaging and video coding has been recently discussed as the basis for a clean-slate approach to low-power wireless video streaming systems based on simple encoder and high-complexity decoder, with applications to wireless multimedia sensor networks [3], [4].

In this context, we propose and study a multi-view video encoding and decoding architecture based on compressive imaging principles, designed to acquire multiple correlated images from the same area of interest from different views. The architecture is motivated by wireless video sensing applications with low-complexity, independent encoders with minimal inter-sensor communication; and a potentially more complex joint decoder, which can lead to substantial rate savings and energy savings on battery-powered wireless sensors. Theoretical results for noiseless [5] and noisy [6] distributed source coding have been available since the late seventies. Several video coding schemes based on CS have been proposed in the literature [4], [7], [8], [9]. However, they mainly focus on performing CS reconstruction by exploiting correlation among successive frames [8], [9] without considering inter-view correlation; or consider rate allocation with traditional CS reconstruction methods [4]. In [10], a distributed multi-view video coding scheme based on CS is proposed, which however assumes the same measurement rates for different views, and can only be applied together with specific structured dictionaries as sparse representation matrix. Differently, in this work we consider multi-view video sequences encoded at different rates and with more general sparsifying matrices, e.g., Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT). The authors of [11] propose a CS-based joint reconstruction method for multi-view images, which uses two images from the two nearest views of the current image (the right and left neighbors) to calculate a prediction frame. However, in our work, only one reference view (not necessarily the nearest one) is selected to reconstruct the side frame for the joint reconstruction process.

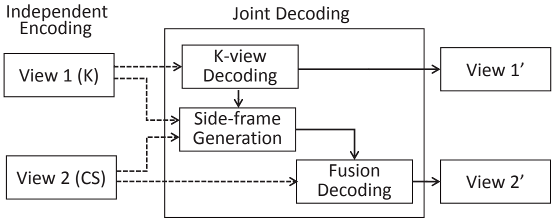


Fig. 1: Muti-view encoding/decoding architecture.

The rest of the paper is organized as follows. In Section II, we introduce the overall encoding/decoding framework, and in Section III, we describe the joint muti-view decoder. We present the simulation results in Section IV, and draw conclusions in Section V.

II. System Architecture

We consider a multi-view video streaming system with N cameras, each capturing a different view of the same scene of interest. Different views are encoded and transmitted independently, and then jointly decoded at the receiver side.

978-1-4799-0294-1/13/$31.00 ©2013 IEEE

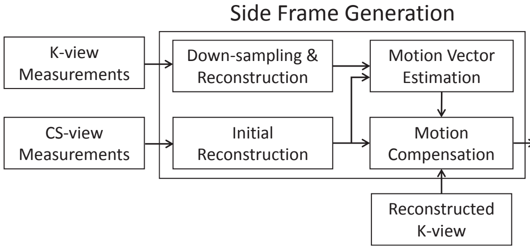


Fig. 2: Block diagram of side frame generation.

Figure 1 illustrates the encoding/decoding architecture, for $N = 2$ - the architecture can be easily extended to $N > 2$.

At the encoder side, one of the considered views is selected as a reference for the other view. We refer to the selected view as *K-view*, and to the other view as *CS-view*. The frames of the K-view and of the CS-view are encoded at a measurement rate of R_{k} and R_{cs}, respectively. We assume that $R_{\mathrm{cs}} \leq R_{\mathrm{k}}$. If we denote by $H \times W$ the dimension of the captured scene (in pixels), then each K-view frame, denoted as $\mathbf{x}_{\mathrm{k}} \in \mathcal{Z}^{H \times W}$ is sampled into a measurement matrix $\mathbf{y}_{\mathrm{k}} \in \mathcal{Z}^{H_{\mathrm{k}} \times W_{\mathrm{k}}}$ with $\frac{H_{\mathrm{k}} \times W_{\mathrm{k}}}{H \times W} = R_{\mathrm{k}}$, and the CS-view frame $\mathbf{x}_{\mathrm{cs}} \in \mathcal{Z}^{H \times W}$ is sampled into $\mathbf{y}_{\mathrm{cs}} \in \mathcal{Z}^{H_{\mathrm{cs}} \times W_{\mathrm{cs}}}$ with $\frac{H_{\mathrm{cs}} \times W_{\mathrm{cs}}}{H \times W} = R_{\mathrm{cs}}$. Readers are referred to [7] for details of the encoding procedure.

At the decoder side, the frames of the K-view are reconstructed based on the received measurements of the K-view only. To reconstruct a CS-view frame, we first generate a side frame based on the received K-view and CS-view measurements. Then, we fuse the initially reconstructed CS-frame with the generated side frame through a newly designed fusion algorithm. In the following, we describe the joint multi-view decoder in detail.

III. Joint Multi-view Decoding

We first reconstruct the frames of the K-view, which will serve as a reference for CS-view frame reconstruction.

A. K-view Decoding

Consider any frame of the K-view video sequence, and denote the *received* measurement vector by $\widehat{\mathbf{y}}_{\mathrm{k}} \in \mathcal{Z}^{H_{\mathrm{k}} \times W_{\mathrm{k}}}$ (i.e., a distorted version of $\mathbf{y}_{\mathrm{k}}$ considering the joint effects of quantization, transmission errors, and packet drops). Then, following CS theory, the K-view frame can be reconstructed by solving a convex optimization problem

$$\begin{aligned} \mathrm{P}_1: \text{minimize} \quad & \|\mathbf{s}\|_1 \\ \text{subject to} \quad & \|\hat{\mathbf{y}}_{\mathrm{k}} - \Phi_{\mathrm{k}}\Psi\mathbf{s}\|_2^2 \leq \epsilon, \end{aligned} \tag{1}$$

and then by mapping $\widehat{\mathbf{x}}_{\mathrm{k}} = \Psi\mathbf{s}^*$, where Φ_{k} and Ψ are the sampling matrix and the sparsifying matrix, respectively, ϵ the predefined error tolerance, and $\mathbf{s}^*$ represents the reconstructed coefficients (i.e., the minimizer of (1)).

B. Side Frame Generation

The core idea behind the proposed technique for generating the side frame is to compensate the reconstructed K-view frame $\widehat{\mathbf{x}}_{\mathrm{k}}$ by estimating the inter-view motion vector. For this purpose, we down-sample the received K-view measurements $\widehat{\mathbf{y}}_k$, and compare the reconstructed lower-quality K-view frame with the initially reconstructed CS-view frame. The block diagram of the side frame generation method is illustrated in Fig. 2.

Initial reconstruction. Denote $\widehat{\mathbf{y}}_{\mathrm{cs}}$ as the received CS-view video measurement, and Φ_{cs} as the corresponding sampling matrix. Then, a *preliminary* reconstructed CS-view frame (denoted by $\widehat{\mathbf{x}}_{\mathrm{cs}}^{\mathrm{p}}$) can be obtained, from $H_{\mathrm{cs}} \times W_{\mathrm{cs}}$ measurements, by substituting $\widehat{\mathbf{y}}_{\mathrm{cs}}$, Φ_{cs} and $\widehat{\mathbf{x}}_{\mathrm{cs}}$ into (1), and by then solving the corresponding optimization problem.

Down-sampling and reconstruction. We then down-sample the received K-view measurement vector $\widehat{\mathbf{y}}_{\mathrm{k}}$ to obtain a new K-view frame with the same (or comparable) reconstructed quality with respect to $\widehat{\mathbf{x}}_{\mathrm{cs}}^{\mathrm{p}}$. Experiments have verified that this leads to more accurate motion vector estimation than using the original K-view frame $\widehat{\mathbf{x}}_{\mathrm{k}}$ reconstructed in Section III-A.

Since $R_{\mathrm{cs}} \leq R_{\mathrm{k}}$ as stated in Section II, without loss of generality, we consider the CS-view sampling matrix Φ_{cs} to be a sub-matrix of Φ_{k}. Then, down-sampling can be achieved by selecting from $\widehat{\mathbf{y}}_{\mathrm{k}}$ only measurements corresponding to Φ_{cs}, which is equivalent, apart from transmission errors and quantization errors, to sampling the original K frame with the matrix used for sampling the CS frame. Denote the down-sampled K-view measurement vector as $\widehat{\mathbf{y}}_{\mathrm{k}}^{\mathrm{d}}$, and the corresponding reconstructed frame with lower quality as $\widehat{\mathbf{x}}_{\mathrm{k}}^{\mathrm{d}}$.

Motion vector estimation. We can now estimate the inter-view motion vector by comparing the preliminarily-reconstructed CS-frame $\widehat{\mathbf{x}}_{\mathrm{cs}}^{\mathrm{p}}$ and the quality-degraded K-frame $\widehat{\mathbf{x}}_{\mathrm{k}}^{\mathrm{d}}$. First, we split $\widehat{\mathbf{x}}_{\mathrm{cs}}^{\mathrm{p}}$ into a set $\mathcal{B}_{\mathrm{cs}}^{\mathrm{p}}$ of blocks with block size $B_{\mathrm{cs}}^{\mathrm{p}} \times B_{\mathrm{cs}}^{\mathrm{p}}$ (in pixel). For each current block $i_{\mathrm{cs}} \in \mathcal{B}_{\mathrm{cs}}^{\mathrm{p}}$, within a predefined search range p in the quality-degraded K-frame $\widehat{\mathbf{x}}_{\mathrm{k}}^{\mathrm{d}}$, a set $\mathcal{B}_{\mathrm{k}}^{\mathrm{d}}(i_{\mathrm{cs}}, p)$ of reference blocks, each with the same block size $B_{\mathrm{cs}}^{\mathrm{p}} \times B_{\mathrm{cs}}^{\mathrm{p}}$, can be identified based on existing strategies [12], e.g., exhaustive search (ES), three step search (TSS), or diamond search (DS). Then, the mean of absolute difference (MAD) between block $i_{\mathrm{cs}} \in \mathcal{B}_{\mathrm{cs}}^{\mathrm{p}}$ and any block $i_{\mathrm{k}} \in \mathcal{B}_{\mathrm{k}}^{\mathrm{d}}(i_{\mathrm{cs}}, p)$ is defined as

$$MAD_{i_{\mathrm{cs}}i_{\mathrm{k}}} = \frac{\sum_{m=1}^{B_{\mathrm{cs}}^{\mathrm{p}}} \sum_{n=1}^{B_{\mathrm{cs}}^{\mathrm{p}}} \left| v_{\mathrm{cs}}^{\mathrm{p}}(i_{\mathrm{cs}}, m, n) - v_{\mathrm{k}}^{\mathrm{d}}(i_{\mathrm{k}}, m, n) \right|}{B_{\mathrm{cs}}^{\mathrm{p}} \times B_{\mathrm{cs}}^{\mathrm{p}}}, \tag{2}$$

with $v_{\mathrm{cs}}^{\mathrm{p}}(i_{\mathrm{cs}}, m, n)$ and $v_{\mathrm{k}}^{\mathrm{d}}(i_{\mathrm{k}}, m, n)$ representing the value of the pixels at (m, n) in block $i_{\mathrm{cs}} \in \mathcal{B}_{\mathrm{cs}}^{\mathrm{p}}$ and $i_{\mathrm{k}} \in \mathcal{B}_{\mathrm{k}}^{\mathrm{d}}(i_{\mathrm{cs}}, p)$, respectively. Let $i_{\mathrm{k}}^* \in \mathcal{B}_{\mathrm{k}}^{\mathrm{d}}(i_{\mathrm{cs}}, p)$ represent the best matching block with the lowest MAD, i.e.,

$$i_{\mathrm{k}}^* = \underset{i_{\mathrm{k}} \in \mathcal{B}_{\mathrm{k}}^{\mathrm{d}}(i_{\mathrm{cs}}, p)}{\arg\min} \; MAD_{i_{\mathrm{cs}}i_{\mathrm{k}}}. \tag{3}$$

and $MAD_{i_{\mathrm{cs}}i_{\mathrm{k}}^*}$ be the corresponding MAD. Furthermore, let $\Delta m(i_{\mathrm{cs}})$ and $\Delta n(i_{\mathrm{cs}})$ represent the horizontal and vertical offset (in pixel) of the block i_{k}^* relative to the current block i_{cs}.

Different from motion vector search in single view encoding [13], for which it is sufficient to search for the block corresponding to the minimum MAD (i.e., block i_{k}^*), in the multi-view case the block i_{k}^* is not necessarily a proper estimation of block i_{cs} due to the possible “hole” problem (i.e., an object that appears in a view is occluded in other views), which can be rather severe.

To address this challenge, we adopt a threshold-based policy. Let $MAD_{\rm th}$ represent the predefined MAD threshold, which can be estimated online by periodically transmitting a frame at a higher measurement rate. Then, if a block $i_{\rm k}^* \in \mathcal{B}_{\rm k}^{\rm d}(i_{\rm cs}, p)$ can be found satisfying $MAD_{i_{\rm cs} i_{\rm k}^*} \leq MAD_{\rm th}$, the current block $i_{\rm cs} \in \mathcal{B}_{\rm cs}^{\rm p}$ is marked as *referenced* with motion vector $(\Delta m(i_{\rm cs}), \Delta n(i_{\rm cs}))$; Otherwise, the block is marked as *non-referenced*.

Motion compensation. The side frame $\mathbf{x}_{\rm si} \in \mathcal{Z}^{H\times W}$ can then be generated by compensating the initially reconstructed CS-view frame $\widehat{\mathbf{x}}_{\rm cs}^{\rm p}$, with given motion vector $(\Delta m(i_{\rm cs}), \Delta n(i_{\rm cs}))$ for each block in $\mathcal{B}_{\rm cs}^{\rm p}$, and the reconstructed K-view frame $\widehat{\mathbf{x}}_{\rm k}$.[1] The compensation works as follows. First, $\mathbf{x}_{\rm si}$ is initialized to $\mathbf{x}_{\rm si} = \widehat{\mathbf{x}}_{\rm cs}^{\rm p}$. Then, we replace each referenced block $i_{\rm cs}$ using the corresponding block from the K-view frame $\widehat{\mathbf{x}}_{\rm k}$ with motion vector $(\Delta m(i_{\rm cs}), \Delta n(i_{\rm cs}))$.

C. Fusion Decoding Algorithm

Finally, we fuse the received CS-view measurements $\widehat{\mathbf{y}}_{\rm cs}$ and the above obtained side-frame $\mathbf{x}_{\rm si}$, in favor of further video quality enhancement, and to remove the block effects of the side frame. This is achieved by generating CS measurements by sampling $\mathbf{x}_{\rm si}$, appending the generated measurements to $\widehat{\mathbf{y}}_{\rm cs}$, and then reconstructing a new CS-view frame based on the combined measurements.

To sample the side frame, we use a sampling matrix Φ, with $\Phi_{\rm cs}$ and $\Phi_{\rm k}$ both being a sub-matrix of Φ. Then, we select a number $R_{\rm si} \times H \times W$ of the resulting measurements, with $R_{\rm si}$ being the predefined measurement rate for the side frame. The value of $R_{\rm si}$ depends on the amount of the CS-view measurement $\widehat{\mathbf{y}}_{\rm cs}$ that have already been received. The larger $R_{\rm cs}$ is, the smaller should be $R_{\rm si}$. No side information is needed in the case of sufficient CS-view measurement. In this work, we empirically set $R_{\rm si}$ as follows:

$$\begin{cases} R_{\rm si} = 1 - R_{\rm cs}, & \text{if } R_{\rm cs} \leq 0.5 \\ R_{\rm si} = 0.6 - R_{\rm cs}, & \text{if } 0.5 < R_{\rm cs} \leq 0.6 \\ R_{\rm si} = 0, & \text{if } R_{\rm cs} > 0.6 \end{cases} \tag{4}$$

D. Blind Video Quality Estimation

Denote $\widehat{\mathbf{x}}_{\rm cs}$ as the final jointly reconstructed CS-view frame. Then, a natural question is: how good is the reconstructed video quality? This is especially critical in CS-based multi-view video streaming systems where the original pixels are not available either at the transmitter or at the receiver side. To address this challenge, we propose a blind video quality estimation method within the coding/decoding framework described above.

First, the reconstructed CS-view frame $\widehat{\mathbf{x}}_{\rm cs}$ is resampled at the CS-view measurement rate $R_{\rm cs}$, with the same sampling matrix $\Phi_{\rm cs}$, thus obtaining $M_{\rm cs}$ new measurements denoted by $\overline{\mathbf{y}}_{\rm cs}$. Then, the PSNR of $\widehat{\mathbf{x}}_{\rm cs}$ with respect to the original frame $\mathbf{x}_{\rm cs}$ (which is not available even at the encoder side) can be estimated as

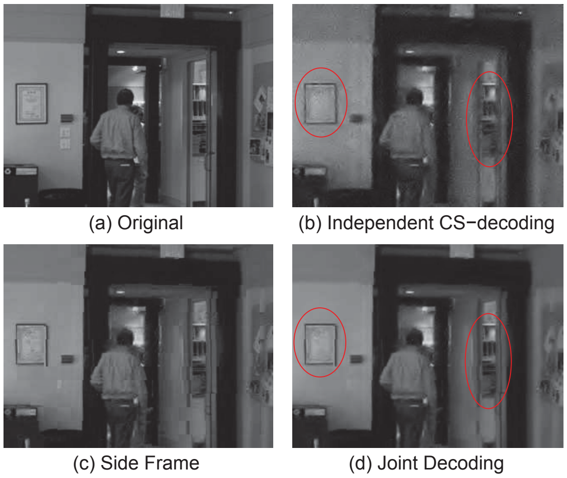


Fig. 3: The fifth frame of *Exit*; Measurement rate is set to 0.2.

$$\text{PSNR} = 10\log_{10}\frac{(2^n-1)^2}{MSE} + \Delta\text{PSNR}, \tag{5}$$

with n being the number of bits per measurement, and

$$\text{MSE} = \frac{\|\,\widehat{\mathbf{y}}_{\rm cs} - \overline{\mathbf{y}}_{\rm cs}\|_2^2}{M_{\rm cs}^2}. \tag{6}$$

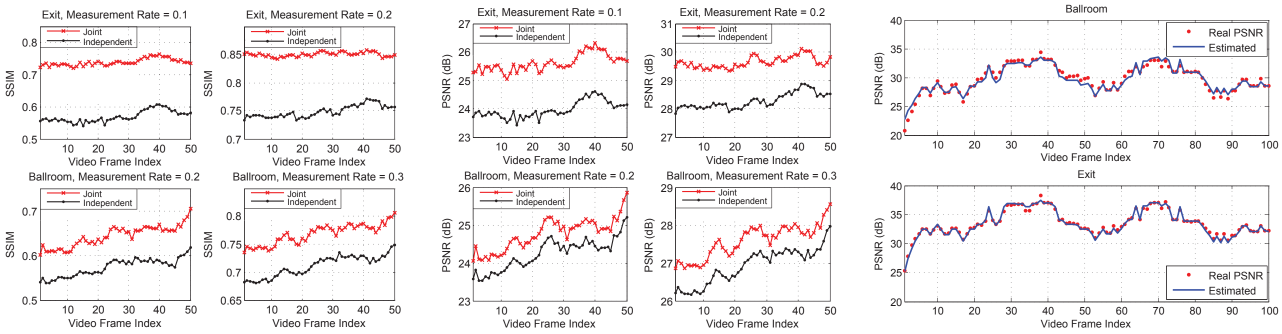
In (5), ΔPSNR is a compensation coefficient that has been found to stay constant or vary only slowly for each view. Hence, it can be estimated online by periodically transmitting a CS-frame at a higher measurement rate.

IV. Simulation Results

We assess the performance of the proposed joint decoder with one K-view and one CS-view. View 1 (defined as K-view with measurement rate 0.6) and View 2 (defined as CS-view with measurement rate 0.1, 0.2, or 0.3) of *Ballroom* and *Exit* multi-view data sets are used in the experiments, representing scenes with fast- and moderate-level movement, respectively. Frames for both views are represented by 8-bit grayscale bitmap, with spatial resolution of 320×240 pixels. At the encoder side, a 32×32 Hadamard matrix is used to generate the sampling matrix $\Phi_{\rm k}$, $\Phi_{\rm cs}$, Φ. TSS [14] is used for motion vector estimation at the decoder side, with block size and search range set to $B = 16$ and $p = 32$, respectively. In the blind video quality estimation algorithm the value of Δ PSNR is set to 6 and 2.9 for *Ballroom* and *Exit*, respectively. GPSR [15] is used to solve P_1 in (1). We compare the video quality of the reconstructed CS-view with that achieved by the independent CS-decoder at a measurement rate 0.1, 0.2, or 0.3.

First, we evaluate the proposed joint decoder considering a specific frame as an example, i.e., the fifth frame of *Exit*. Results are reported in Fig. 3. We found that the blurring effect in the independently reconstructed frame is mitigated through joint decoding. Taking the regions within ellipses in Fig. 3(b) and (d) as example, we can see that the video quality improvement is noticeable, which corresponds to a Structural Similarity (SSIM) [16] improvement of 0.11 (from 0.74 to

[1] Note that we estimate the motion vector based on the quality-degraded K-view frame, but compensate the initially reconstructed CS-view frame using the K-view frame at the original quality.



(a) (b) (c)

Fig. 4: Simulation results for CS-view: (a) SSIM comparison for *Exit* and *Ballroom*, (b) PSNR comparison for *Exit* and *Ballroom*, (c) Evaluation of the blind quality estimation algorithm.

0.85). The block effect in the side frame in Fig. 3(c) has also been removed.

Then, we compare the achieved SSIM and PSNR of (i) CS-view at a measurement rate 0.1, 0.2 and 0.3 with independent decoding (in black), and (ii) with our proposed joint decoding method (in red) for the first 50 frames of *Exit* and *Ballroom*. This is illustrated in Fig. 4(a) and 4(b), respectively, which show that the proposed algorithm outperforms independent CS decoding up to 0.12 in terms of SSIM and 1.6 dB in terms of PSNR with both low-motion *Exit* and fast-motion *Ballroom* sequences, and at different measurement rates.

Finally, to evaluate the proposed blind quality estimation method, we transmit the CS-view sequence over time-varying channels with a randomly generated error pattern. The K-view is assumed to be correctly received and reconstructed (e.g., through strong channel error protection). A setting similar to [4] is considered for CS-view transmission, i.e., the encoded CS-view measurements are first quantized and packetized. Then, parity bits are added to each packet. A packet is dropped at the receiver if detected to contain errors after a parity check. The result for 100 successive frames is illustrated in Fig. 4(c) (where the top figure refers to *Ballroom*, while the bottom refers to *Exit*), between real PSNR (red dot) and estimated PSNR (blue line). We can conclude that our proposed blind estimation within our joint decoding of independently encoding framework is rather precise, with an estimation error of 4.32% for *Ballroom* and of 6.50% for *Exit*, respectively.

V. Conclusions

We have designed a new scheme for jointly decoding independently- and compressively-sampled multi-view video streams. Simulation results showed that the proposed joint decoder outperforms the independent CS-decoder in the case of both fast and moderate motion levels. The accuracy of a newly-proposed blind video quality estimation method was also verified.

Acknowledgement

This paper is based upon work supported in part by the US National Science Foundation under grant CNS1117121 and by the US Office of Naval Research under grant N00014-11-1-0848. Zhangyu Guan's work is also supported in part by the China Postdoctoral Science Foundation under grant 2012M521332 and by the Shandong Doctoral Foundation under grant 2012BSE27052.

References

[1] E. J. Candes and M. B. Wakin, “An Introduction to Compressive Sampling,” *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, March 2008.

[2] D. L. Donoho, “Compressed Sensing,” *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, April 2006.

[3] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, “A Survey on Wireless Multimedia Sensor Networks,” *Computer Networks*, vol. 51, no. 4, pp. 921–960, March 2007.

[4] S. Pudlewski, T. Melodia, and A. Prasanna, “Compressed-sensing Enabled Video Streaming for Wireless Multimedia Sensor Networks,” *IEEE Transactions on Mobile Computing*, vol. 11, no. 6, pp. 1060–1072, June 2012.

[5] D. Slepian and J. K. Wolf, “Noiseless Coding of Correlated Information Sources,” *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 471–480, July 1973.

[6] A. D. Wyner and J. Ziv, “The Rate-distortion Function for Source Coding with Side-information at the Decoder,” *IEEE Transactions on Information Theory*, vol. 22, no. 1, pp. 1–10, Jan. 1976.

[7] S. Pudlewski and T. Melodia, “A Tutorial on Encoding and Wireless Transmission of Compressively Sampled Videos,” *IEEE Communications Surveys & Tutorials*, vol. 15, no. 2, pp. 754–767, Second Quarter 2013.

[8] H. W. Chen, L. W. Kang, and C. S. Lu, “Dynamic Measurement Rate Allocation for Distributed Compressive Video Sensing,” *Visual Communications and Image Processing*, vol. 7744, pp. 1–10, July 2010.

[9] Y. Liu, M. Li, and D. A. Pados, “Motion-aware Decoding of Compressed-sensed Video,” *IEEE Transactions on Circuits System Video Technology*, vol. 23, no. 3, pp. 438–444, March 2013.

[10] X. Chen and P. Frossard, “Joint Reconstruction of Compressed Multi-view Images,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Taipei, Taiwan, April 2009.

[11] M. Trocan, T. Maugey, E. W. Tramel, J. E. Fowler, and B. Pesquet-Popescu, “Compressed Sensing of Multiview Images Using Disparity Compensation,” in *Proc. IEEE International Conference on Image Processing (ICIP)*, pp. 3345–3348, Hong Kong, Sep. 2010.

[12] F. H. Jamil, R. R. Porle, A. Chekima, R. A. Lee, H. Ali, and S. M. Rasat, “Preliminary Study of Block Matching Algorithm (BMA) for Video Coding,” in *Proc. International Conference On Mechatronics (ICOM)*, Istanbul, Turkey, May 2011.

[13] A. M. Huang and T. Nguyen, “Motion Vector Processing Using Bidirectional Frame Difference in Motion Compensated Frame Interpolation,” in *Proc. IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks*, Newport Beach, CA, USA, June 2008.

[14] T. Koya, K. Lunuma, A. Hirano, Y. Lyima, and T. Ishi-guro, “Motion-compensated Inter-frame Coding for Video Conferencing,” in *Proc. National Telecommunications Conference (NTC)*, New Orleans, LA, USA, Nov. 1981.

[15] 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 on Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 586–598, Dec. 2007.

[16] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, “Image Quality Assessment: From Error Visibility to Structural Similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, April 2004.