Multi-view Wireless Video Streaming Based on Compressed Sensing: Architecture and Network Optimization

Nan Cen[†], Zhangyu Guan[†‡], Tommaso Melodia[†]
[†]Department of Electrical and Computer Engineering
Northeastern University, Boston, MA 02115
[‡]Department of Electrical Engineering
State University of New York at Buffalo, Buffalo, NY 14226
{ncen, zguan, melodia}@ece.neu.edu

ABSTRACT

Multi-view wireless video streaming has the potential to enable a new generation of efficient and low-power pervasive surveillance systems that can capture scenes of interest from multiple perspectives, at higher resolution, and with lower energy consumption. However, state-of-the-art multi-view coding architectures require relatively complex predictive encoders, thus resulting in high processing complexity and power requirements. To address these challenges, we consider a wireless video surveillance scenario and propose a new encoding and decoding architecture for multi-view video systems based on Compressed Sensing (CS) principles, composed of cooperative sparsity-aware block-level rate-adaptive encoders, feedback channels and independent decoders. The proposed architecture leverages the properties of CS to overcome many limitations of traditional encoding techniques, specifically massive storage requirements and high computational complexity. It also uses estimates of image sparsity to perform efficient rate adaptation and effectively exploit inter-view correlation at the encoder side.

Based on the proposed encoding/decoding architecture, we further develop a CS-based end-to-end rate distortion model by considering the effect of packet losses on the perceived video quality. We then introduce a modeling framework to design network optimization problems in a multi-hop wireless sensor network. Extensive performance evaluation results show that the proposed coding framework and power-minimizing delivery scheme are able to transmit multi-view streams with guaranteed video quality at low power consumption.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication*

Keywords

Compressed sensing; Multi-view video streaming; Network optimization.

MobiHoc'15, June 22–25, 2015, Hangzhou, China.

http://dx.doi.org/10.1145/2746285.2746309.

1. INTRODUCTION

Low-power wireless video monitoring and surveillance systems (sometimes referred to as wireless multimedia sensor networks (WMSNs) [1]) have the potential to enable new generations of monitoring and surveillance systems, e.g., multi-view surveillance networks composed of wirelessly interconnected low-power and low-complexity sensing devices equipped with audio and visual information collection modules that are able to ubiquitously capture multimedia content from environments of interest. In wireless multi-view video streaming systems, arrays of miniature camera sensors simultaneously capture scenes from different perspectives [2] and then deliver the captured video data to the decoder through wireless links. This often comes with large storage requirements and intense processing loads.

While there has been intense research and considerable progress in wireless video sensing systems, *how to enable real-time quality-aware power-efficient multi-view video streaming in large-scale, possibly multi-hop, wireless networks of battery-powered embedded devices is still a substantially open problem.* State-of-the-art Multi-view Video Coding (MVC) technologies such as MVC H.264/AVC [3,4] are mainly based on predictive encoding techniques, i.e., selecting one frame (referred to as reference frame) in one view (referred to as reference view), based on which they perform motion compensation and disparity compensation to predict other intra-view and inter-view frames, respectively. As a consequence, they are characterized by the following fundamental limitations when applied to multi-view streaming in multi-hop wireless sensor networks:

Large storage space, high power consumption and encoder complexity on embedded devices. State-of-the-art MVC technologies incorporating inter-view and intra-view prediction require extra storage space for reference views and frames. They also induce intensive computational complexity at the encoder, which further results in high processing load or additional cost for specialized processors (to perform operations such as motion estimation and compensation) and high power consumption.

Prediction-based encoding techniques are vulnerable to channel errors. In predictive encoding approaches, errors in independently encoded frames can lead to error propagation on the predictively encoded frames, which is especially detrimental in wireless networks with lossy links, where best-effort delivery scheme with simple error detection schemes such as UDP are usually adopted [5]. Therefore, to guarantee multi-view video streaming quality, a desirable

MVC framework should allow graceful degradation of video quality as the channel quality decreases.

Recently, so-called compressed sensing (CS) techniques have been proposed that are able to reconstruct image or video signals from a relatively "small" number of (random or deterministic) linear combinations of original image pixels, referred to as measurements, *without collecting the entire frame* [6, 7], thereby offering a promising alternative to traditional video encoders by *acquiring and compressing video or images simultaneously at very low computational complexity for encoders* [8]. This attractive feature motivated a number of works that have applied CS to video streaming in low-power wireless surveillance scenarios. For example, [9–11] mainly concentrate on single-view CS-based video compression, by exploiting temporal correlation among successive video frames [9, 10] or considering energy-efficient rate allocation in WMSNs with traditional CS reconstruction methods [11]. In [12], we showed that CS-based wireless video streaming can deliver surveillance-grade video for a fraction of the energy consumption of traditional systems based on predictive video encoding such as H.264. In addition, [11] illustrated and evaluated the *error-resilience* property of CS-based video streaming, which results in graceful quality degradation in wireless lossy links. A few recent contributions [13–15] have proposed CS-based multi-view video streaming techniques, primarily focusing on an independent-encoder and joint-decoder paradigm, which exploits the implicit correlation among multiple views at the decoder side to improve the resulting video quality using complex joint reconstruction algorithms.

From a systems perspective, how to allocate power-efficient rates to different views for a required level of video quality is another important open problem in wirelessly networked multi-view video streaming systems. Very few algorithms have been reported in the literature to address this issue. For example, [16] and [17] have looked at this problem by considering traditional encoding paradigms, e.g., H.264 or MPEG4; these contributions focus on video transmission in single-hop wireless networks and provide a framework to improve power efficiency by adjusting encoding parameters such as quantization step (QS) size to adapt the resulting rate.

To bridge the aforementioned gaps, in this paper we first propose a novel CS-based multi-view coding and decoding architecture composed of cooperative encoders and independent decoders. Unlike existing works [13–15], the proposed system is based on independent encoding and independent decoding procedures with limited channel feedback information and negligible content sharing among camera sensors. Furthermore, we propose a power-efficient quality-guaranteed rate allocation algorithm based on a compressive Rate-Distortion (R-D) model for multi-view video streaming in multi-path multi-hop wireless sensor networks with lossy links. Our work makes the following contributions:

CS-based multi-view video coding architecture with independent encoders and independent decoders. Different from state-of-the-art multi-view coding architectures, that are either based on joint encoding or on joint decoding, we propose a new CS-based sparsity-aware independent encoding and decoding multi-view structure, that relies on lightweight feedback and inter-camera cooperation.

- *Sparsity estimation.* We develop a novel adaptive approach to estimate block sparsity based on the reconstructed frame at the decoder. The estimated sparsity is then used to calculate the block-level measurement rate to be allocated with respect to a given frame-level rate. Next, the resulting block-level rates are transmitted back to the encoder through the feedback channel. The encoder that is selected to receive the feedback information, referred to as reference view (R-view), shares the content with other non-reference views (NR-views) nearby.

- *Block-level rate adaptive multi-view encoders.* R-view and NR-views perform the block-level CS encoding independently based on the shared block-level measurement rate information. The objective is to not only implicitly leverage the considerable correlation among views, but also to adaptively balance the number of measurements among blocks with different sparsity levels. Our experimental results show that the proposed method outperforms state-of-the-art CS-based encoders with equal block-level measurement rate by up to 5 dB.

Modeling framework for CS-based multi-view video streaming in multi-path multi-hop wireless sensor networks. We consider a rate-distortion model of the proposed streaming system that captures packet losses caused by unreliable links and playout deadline violations. Based on this model, we propose a two-fold (frame-level and path-level) rate control algorithm designed to minimize the network power consumption under constraints on the minimum required video quality for multi-path multi-hop multi-view video streaming scenarios.

The rest of the paper is organized as follows. In Section 2, we review a few preliminary notions. In Section 3, we introduce the proposed CS-based multi-view video encoding/decoding architecture. In Section 4, we discuss the modified R-D model, and in Section 5 we present a modeling framework to design optimization problems of multi-view streaming in multi-hop sensor networks based on the end-to-end R-D model. Finally, simulation results are presented in Section 6, while in Section 7 we draw the main conclusions and discuss future work.

2. PRELIMINARIES

2.1 Compressed Sensing Basics

We first briefly review basic concepts of CS for signal acquisition and recovery, especially as applied to CS-based video streaming. We consider an image signal vectorized and then represented as $\mathbf{x} \in \mathbb{R}^N$, where $N = H \times W$ is the number of pixels in the image, and H and W represent the dimensions of the captured scene. Each element x_i denotes the i^{th} pixel in the vectorized image signal representation. Most natural images are known to be very nearly sparse when represented using some transformation basis $\mathbf{\Psi} \in \mathbb{R}^{N \times N}$, e.g., Discrete Wavelet Transform (DWT) or Discrete Cosine Transform (DCT), denoted as $\mathbf{x} = \mathbf{\Psi s}$, where $\mathbf{s} \in \mathbb{R}^N$ is sparse representation of $\mathbf{x}$. If $\mathbf{s}$ has at most K nonzero components, we call $\mathbf{x}$ a K-sparse signal with respect to $\mathbf{\Psi}$.

In CS-based imaging system, sampling and compression are executed simultaneously through a linear measurement matrix $\mathbf{\Phi} \in \mathbb{R}^{M \times N}$, with $M \ll N$, as

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

with $\mathbf{y} \in \mathbb{R}^M$ representing the resulting sampled and compressed vector.

It was proven in [6] that if $\mathbf{A} \triangleq \mathbf{\Phi\Psi}$ satisfies the following Restricted Isometry Property (RIP) of order K,

$$(1-\delta_k)||\mathbf{s}||_{l_2}^2 \leq ||\mathbf{As}||_{l_2}^2 \leq (1+\delta_k)||\mathbf{s}||_{l_2}^2, \quad (2)$$

with $0 < \delta_k < 1$ being a small "isometry" constant, then we can recover the optimal sparse representation $\mathbf{s}^*$ of $\mathbf{x}$ by solving the following optimization problem

$$\begin{aligned} \mathrm{P}_1: \quad & \text{Minimize} \quad ||\mathbf{s}||_0 \\ & \text{Subject to:} \quad \mathbf{y} = \mathbf{\Phi\Psi s} \end{aligned} \quad (3)$$

by taking only

$$M = c \cdot K\log(N/K) \quad (4)$$

measurements, where c is some predefined constant. Afterwards, $\mathbf{x}$ can be obtained by

$$\hat{\mathbf{x}} = \mathbf{\Psi s}^*. \quad (5)$$

However, problem P_1 is NP-hard in general, and in most practical cases, measurements $\mathbf{y}$ may be corrupted by noise, e.g., channel noise or quantization noise. Then, most state-of-the-art work relies on l_1 minimization with relaxed constraints in the form

$$\begin{aligned} \mathrm{P}_2: \quad & \text{Minimize} \quad ||\mathbf{s}||_1 \\ & \text{Subject to :} \quad ||\mathbf{y} - \mathbf{\Phi\Psi s}||_2 \leq \epsilon \end{aligned} \quad (6)$$

to recover $\mathbf{s}$. Note that P_2 is a convex optimization problem. Researchers in sparse signal reconstruction have developed various solvers [18–20]. For example, the Least Absolute Shrinkage and Selection Operator (LASSO) solver [19] can solve problem P_2 with computational complexity $\mathcal{O}(M^2N)$. We consider a Gaussian random measurement matrix $\mathbf{\Phi}$ in this paper.

2.2 Rate-Distortion Model for Compressive Imaging

Throughout this paper, end-to-end video distortion is measured as mean squared error (MSE). Since Peak Signal-to-Noise Ratio (PSNR) is a more common metric in the video coding community, we use $\mathrm{PSNR} = 10\log_{10}(255^2/\mathrm{MSE})$ to illustrate simulation results. The distortion at the decoder D_{dec} in general includes two terms, i.e., D_{enc}, distortion introduced by the encoder (e.g., not enough measurements and quantization); and D_{loss}, distortion caused by packet losses due to unreliable wireless links and violating playout deadlines because of bandwidth fluctuations. Therefore,

$$D_{\mathrm{dec}} = D_{\mathrm{enc}} + D_{\mathrm{loss}}. \quad (7)$$

To the best of our knowledge, there are only a few works [11] that have investigated rate-distortion models for compressive video streaming, but without considering losses. For example, [11] expands the distortion model in [21] to CS video transmission as

$$D(R) = D_0 + \frac{\theta}{R - R_0}, \quad (8)$$

where D_0, θ and R_0 are image- or video-dependent constants that can be determined by linear least squares fitting techniques; $R = \frac{M}{N}$ is the user-controlled measurement rate of each video frame.

3. CS-BASED MULTI-VIEW CODING ARCHITECTURE DESIGN

In this section, we introduce a novel encoding/decoding architecture design for CS multi-view video streaming. The proposed framework is based on three main components: (i) cooperative sparsity-aware block-level rate adaptive encoder, (ii) independent decoder, and (iii) a centralized controller located at the decoder. As illustrated in Fig. 1, considering a two-view example, camera sensors acquire a scene of interest with adaptive block-level rates and transmit sampled measurements to the base station/controller through a multi-path multi-hop wireless sensor network. Then, the centralized controller calculates the relevant information and feeds it back to the selected R-view. The R-view then shares the limited feedback information with the other one - NR-view. The architecture can be easily extended to $V \geq 2$ views.

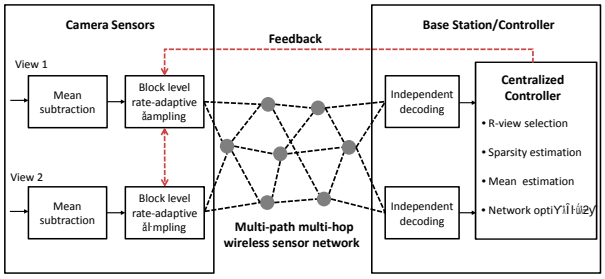


Figure 1: Encoding/decoding architecture for multi-hop CS-based multi-view video streaming.

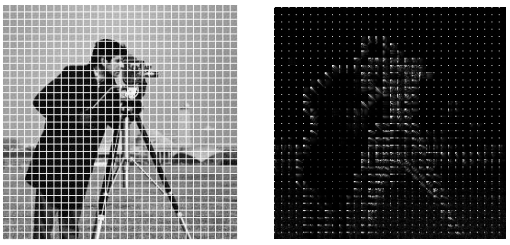
Different from existing compressive encoders with equal block measurement rate [10, 11], the objective of the proposed framework is to improve the reconstruction quality by leveraging each block's sparsity as a guideline to adapt the block-level measurement rate. We next describe how to implement the proposed paradigm by discussing each component in detail.

3.1 Cooperative Block-level Rate-adaptive Encoder

To reduce the computational burden at encoders embedded in power-constrained devices, most state-of-the-art multi-view proposals focus on developing complex joint reconstruction algorithms to improve the reconstruction quality. Differently, in our architecture we obtain improved quality only through sparsity-aware encoders.

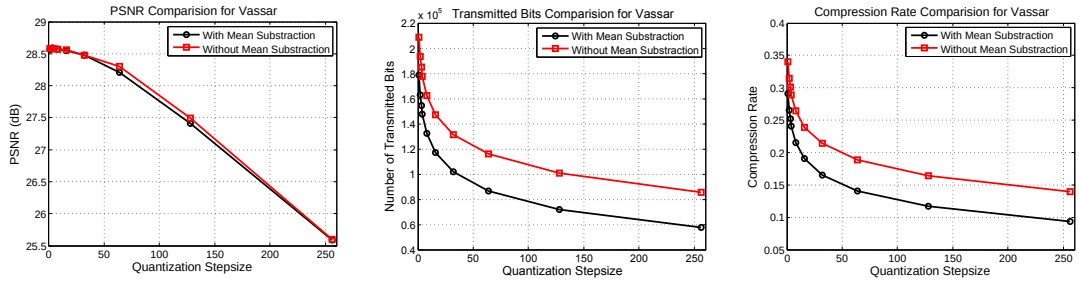
To illustrate the idea, Figure 2(b) depicts the sparse representation of Fig. 2(a) with respect to block-based DCT transformation. We can observe that sparsity differs among blocks, e.g., the blocks within the coat area are more sparse than others. According to basic compressed sensing theory in Section 2.1, (4) indicates that the number of required measurements is inversely proportional to the sparsity K. Therefore, we propose to adapt the measurement rate at the block level according to sparsity information, i.e., more measurements will be allocated to less-sparse blocks, and vice versa.

In our work, the number of required measurements M_{vf}^i for block i in frame f of view v, $1 \leq i \leq B$, is calculated based on the sparsity estimated at the centralized controller and sent back via a feedback channel. Here, $B = \frac{N}{N_b}$ denotes



(a) (b)

Figure 2: Block Sparsity: (a) Original image, (b) Block-based DCT coefficients of (a).



(a) (b) (c)

Figure 3: Comparison of (a) PSNR, (b) the number of transmitted bits, and (c) the compression rate between approaches with and without mean subtraction.

the total number of blocks in one frame with N and N_b being the total number of pixels in one frame and block, respectively. Assume that we have received $\{M^i_{vf}\}_{i=1}^B$. Then, the encoding process is similar to (1), described as

$$\mathbf{y}^i_{vf} = \mathbf{\Phi}^i_{vf}\mathbf{x}^i_{vf}, \tag{9}$$

where $\mathbf{y}^i_{vf} \in \mathbb{R}^{M^i_{vf}}$ and $\mathbf{\Phi}^i_{vf} \in \mathbb{R}^{M^i_{vf}\times N_b}$ are the measurement vector and measurement matrix for block i in frame f of view v, respectively; $\mathbf{x}^i_{vf} \in \mathbb{R}^{N_b}$ represents the original pixel vector of block i. From (9), we can see that M^i_{vf} varies among blocks from 1 to N_b, thereby implementing block-level rate adaptation. In Section 6, the simulation results will show that this approach can improve the quality by up to 5 dB.

Mean value subtraction. The CS-based imaging system acquires and compresses each frame simultaneously through simple linear operations as in (1). Therefore, it can help reduce the energy consumption compared with traditional signal acquisition and encoding approaches (e.g., H.264/AVC) that are based on complicated motion estimation and motion compensation operations. However, the compression rate of CS is not as high as traditional encoding schemes [12]. To compensate for this, we perform *mean value subtraction*, which can further help reduce the number of transmitted bits. How to obtain the mean value $\bar{m}$ will be discussed in Section 3.3. Since the original pixels are not available at the compressive encoder, we perform the *mean value subtraction in the measurement domain*. First, we establish a mean value vector $\mathbf{m} \in \mathbb{R}^{N_b}$ with dimensions the same as $\mathbf{x}^i_{vf}$, and where each element is equal to $\bar{m}$. Then, we use the same block-level measurement matrix $\mathbf{\Phi}^i_{vf}$ to sample $\mathbf{m}$ and then subtract the result from $\mathbf{y}^i_{vf}$ as

$$\tilde{\mathbf{y}}^i_{vf} = \mathbf{y}^i_{vf} - \mathbf{\Phi}^i_{vf}\mathbf{m} = \mathbf{\Phi}^i_{vf}(\mathbf{x}^i_{vf} - \mathbf{m}). \tag{10}$$

After sampling, $\tilde{\mathbf{y}}^i_{vf}$ is transmitted to the decoder. From (10), we can see that the proposed mean value subtraction in the measurement domain is equivalent to subtraction in the pixel domain.

Next, to validate the effectiveness of mean value subtraction, we take the *Vassar* sequence as an example. We select a uniform quantization method. The forward quantization stage and the reconstruction stage can be expressed as $q = \text{sgn}(x)\cdot\lfloor\frac{|x|}{\Delta}+\frac{1}{2}\rfloor$ and $\hat{q} = \Delta\cdot q$, respectively. Here, x, q, $\hat{q}$ and Δ represent original signal, quantized signal, de-quantized signal and quantization step size, respectively. Figure 3 shows a comparison of PSNR, the number of transmitted bits and the compression rate with and without mean subtraction, where a measurement rate 0.2 is used, and the total bits in the original frame are $320\times240\times8 = 614400$ bits. Quantization step sizes from the set {1, 2, 3, 4, 8, 16, 32, 64, 128, 256} are selected. From Fig. 3(a), we can observe that mean subtraction has a negligible effect on the reconstruction quality and there is no significant quality degradation when the quantization step size is less than 32. This is because the value of measurement is up to thousand and tens of thousand compared to original pixel value with maximum 255. Figures 3(b) and (c) illustrate that with mean subtraction the total number of bits transmitted for one frame is significantly reduced by up to 30 kbits compared to not using mean subtraction, which corresponds to an improvement in compression rate from 0.2391 to 0.1902.

Cooperation via sparsity pattern sharing. Multi-view video streaming is based on reducing the redundancy among views captured by arrays of camera sensors that are assumed to be close enough to each other. Most state-of-the-art literature adopts the concept of distributed system coding architecture [22, 23], where a reference view transmits more measurements than other non-reference views and then the receiver jointly decodes by exploiting the implicit correlation among views. Instead, we allow the encoders to explicitly cooperate to a certain extent. For example, the R-view selected by the centralized controller will periodically receive feedback information, i.e., $\{M_i\}_{i=1}^B$ and $\bar{m}$, and then share it with the NR-views in the same group. Since camera sensors in the same group are assumed to be close enough to each other, the block sparsity among views will be correlated. By using the same sparsity information, we can directly exploit multi-view correlation at the encoders, thus resulting in a clean-slate compressive multi-view coding framework with simple encoders and simple decoders but with improved reconstruction quality.

3.2 Independent Decoder

As mentioned above, the proposed framework results in relatively simple decoders. At each decoder, the received $\hat{\mathbf{y}}^i_{vf}$, distorted version of $\tilde{\mathbf{y}}^i_{vf}$ because of the joint effects of quantization, transmission errors, and packet drops, will be independently decoded. The optimal solution $\mathbf{s}^{i,\star}_{vf}$ can be obtained by solving

$$\begin{aligned} \text{P}_3: \quad &\text{Minimize} \quad ||\mathbf{s}^i_{vf}||_1 \\ &\text{Subject to:} \quad ||\hat{\mathbf{y}}^i_{vf} - \mathbf{\Phi}^{\mathbf{i}}_{\mathbf{vf}}\mathbf{\Psi}_b\mathbf{s}^i_{vf}||_2 \le \epsilon, \end{aligned} \tag{11}$$

where $\mathbf{\Psi}_b \in \mathbb{R}^{N_b\times N_b}$ represents the sparsifying matrix (2-D DCT in this work). We then use (5) to obtain the reconstructed block-level image $\hat{\mathbf{x}}^i_{vf}$, by solving $\hat{\mathbf{x}}^i_{vf} = \mathbf{\Psi}_b\mathbf{s}^{i,\star}_{vf}$. Afterward, $\{\hat{\mathbf{x}}^i_{vf}\}_{i=1}^B$ can be simply reorganized to obtain the reconstructed frame $\hat{\mathbf{x}}_{vf}$.

3.3 Centralized Controller

The centralized controller is the key component at the receiver, which is mainly in charge of selecting the R-view and estimating sparsity and mean value required to be sent back to the transmitter. Additionally, the controller is also responsible for implementing the power-efficient multi-path rate allocation algorithm discussed in Section 5. Next, we

introduce the three key functions executed at the controller in sequence, i.e., *R-view selection, sparsity estimation, and mean value estimation.*

R-view selection. The controller selects a view to be used as reference view (R-view) among views in the same group and then sends feedback information to the selected R-view. For this purpose, the controller first calculates the Pearson correlation coefficients among the measurement vectors of any two views as

$$\rho_{mn} = \text{corr}(\hat{\mathbf{y}}_{mf}, \hat{\mathbf{y}}_{nf}),\ \forall m \neq n,\ m,n = 1,\ldots,V, \quad (12)$$

where $\hat{\mathbf{y}}_{mf}$ is the simple cascaded version of all $\hat{\mathbf{y}}^i_{mf}$ and $\text{corr}(\hat{\mathbf{y}}_{mf}, \hat{\mathbf{y}}_{nf}) \triangleq \frac{\text{cov}(\hat{\mathbf{y}}_{mf}, \hat{\mathbf{y}}_{nf})}{\sigma_{mf}\sigma_{nf}}$. Then, view $m^\star$, referred to as R-view, is selected by solving

$$m^\star = \underset{m=1,\ldots,V}{\text{argmax}}\, \tilde{\rho}_m, \quad (13)$$

where $\tilde{\rho}_m \triangleq \frac{1}{V-1}\sum_{n\neq m}\rho_{mn}$. The reconstructed frame $\hat{\mathbf{x}}_{vf}$ of the R-view is then used to estimate the block sparsity K^i and the frame mean value $\bar{m}$ for block i.

Next, we take the *Vassar* 5-view scenarios as an example, Table 1 shows the calculated $\tilde{\rho}_m$. We can see that the average Pearson correlation coefficient of view 3 is the largest. Therefore, view 3 is selected as R-view. Moreover, to elaborate how much quality gain we can obtain if the other views except view 3 are selected as R-view, we also set them as R-view and calculate the average improved PSNR, respectively, as shown in Table 2. We can observe that the improved average PSNR is proportional to $\tilde{\rho_m}$, where selecting view 3 as R-view results in the highest improved average PSNR gain, i.e., 1.6674 dB. For this case, because the *Vassar* multi-view sequences used here is captured by parallel-deployed cameras with equal spacing, we obtain the same result, i.e., view 3 as R-view, as if we were to choose simply the most central sensor. However, for scenarios with cameras that are not parallel-deployed with unequal spacing, selecting the most central sensor is not necessarily a good choice.

Table 1: Average Pearson correlation coefficient for *Vassar* five views.

	View 1	View 2	View 3	View 4	View 5
$\tilde{\rho}_m$	0.8184	0.8988	0.9243	0.8973	0.8435

Table 2: Improved average PSNR (dB) when selecting different *Vassar* views as R-view.

R-view	View 1	View 2	View 3	View 4	View 5
PSNR (dB)	1.2312	1.6241	1.6674	1.6167	1.3833

Sparsity estimation. Since the original frame in the pixel domain is not available, we propose to estimate sparsity based on the reconstructed frame $\hat{\mathbf{x}}_{vf}$ as follows. By solving the optimization problem P_3 in (11), we can obtain the block sparse representation $\mathbf{s}^{i,\star}_{vf}$ and then reorganize $\{\mathbf{s}^{i,\star}_{vf}\}^B_{i=1}$ to get the frame sparse representation $\mathbf{s}^\star_{vf}$ periodically. The sparsity coefficient K^i is defined as the number of non-zero entries of $\mathbf{s}^\star_{vf}$. However, natural pictures in general are not exactly sparse in the transform domain. Hence, we introduce a predefined percentile p_s, and assume that the frame can be perfectly recovered with $N \cdot p_s$ measurements. Based on this, one can adaptively find a threshold T above which transform-domain coefficients are considered as non-zero entries. The threshold can be found by solving

$$\frac{||\max(|s^\star_{vf}| - T, 0)||_0}{N} = p_s. \quad (14)$$

Then, we apply T to each block i to estimate the block sparsity K^i as

$$K^i = ||\max(|s^{i,\star}_{vf}| - T, 0)||_0. \quad (15)$$

According to (4) and given the frame measurement rate R, M^i_{vf} can then be obtained as

$$M^i_{vf} = \frac{K^i \log_{10}(\frac{N_b}{K_i})}{\sum_{i=1}^B K^i \log_{10}(\frac{N_b}{K_i})} NR. \quad (16)$$

Mean value estimation. Finally, the mean value $\bar{m}$ can be estimated from $\hat{\mathbf{x}}_{vf}$ as

$$\bar{m} = \frac{1}{N}\sum_{i=1}^N \hat{\mathbf{x}}_{vf}(i). \quad (17)$$

With limited feedback and lightweight information sharing, implementing block-level rate adaptation at the encoder without adding computational complexity can improve the reconstruction performance of our proposed encoding/decoding paradigm. This claim will be validated in Section 6 in terms of Peak Signal-to-Noise Ratio (PSNR) and Structure Similarity (SSIM) [24].

4. END-TO-END RATE-DISTORTION MODEL

To handle CS-based multi-view video streaming with guaranteed quality, a rate-distortion model to measure the end-to-end distortion that jointly captures the effects of encoder distortion and transmission distortion as stated in (7) is needed. To this end, we modify the R-D model (8) proposed in [11] by adding a packet loss term to jointly account for compression loss and packet loss in compressive video wireless streaming systems. In traditional predictive-encoding based imaging systems, the importance of packets is not equal (i.e., I-frame packets have higher impact than P-frame and B-frame packets on the reconstructed quality). Instead, each packet in CS-based imaging systems has the same importance, i.e., it contributes equally to the reconstruction quality. Therefore, the packet loss probability p_{loss} can be converted into a measurement rate reduction through a conversion parameter κ and considered into the rate-distortion performance, described as

$$D_{\text{dec}} = D_{\text{enc}} + D_{\text{loss}} = D_0 - \frac{\theta}{R - \kappa p_{\text{loss}} - R_0}. \quad (18)$$

However, how to derive captured-scene-dependent constants D_0, θ, and R_0 in (18) is not trivial. The reasons are listed as follows:

1) Packet loss rate plays a fundamental role in the modified R-D model. In multi-view video streaming in multi-path multi-hop wireless network, how to model the packet loss rate as accurately as possible is still an open problem. In Section 5, we describe our proposed packet loss probability model in detail.

2) The original pixel values are not available at the receiver

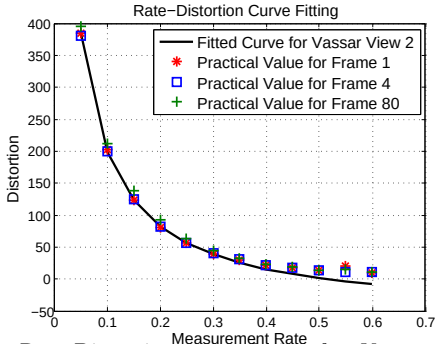


Figure 4: Rate-Distortion curve fitting for *Vassar* view 2 sequence.

end and even not available at the transmitter side in compressive multi-view streaming systems. To address this challenge, we develop a simple but very effective online estimation approach to obtain these three fitting parameters. We let the R-view periodically transmit a frame at a higher measurement rate, e.g., 60% measurement, and after reconstruction at the decoder side, the reconstructed frame is considered as the original image in the pixel domain. We then resample it at different measurement rates and perform the reconstruction procedure again. Finally, approximate distortion in terms of MSE can be calculated between the reconstructed frame at lower measurement rates and the reconstructed frame with 60% measurements.

We take the *Vassar* view 2 sequence as example. According to the above-mentioned online rate-distortion estimation approach, a measurement rate of 0.6 is selected.. Figure 4 illustrates the simulation results, where the black solid line is the rate-distortion curve fitted through a linear least-square approach. To evaluate this approach, we calculate the distortion value for frames 1, 4 and 80 at different measurement rates and then compare them with the estimated rate-distortion curve, where ground-truth distortion values are depicted as red pentagrams, blue squares and green pluses compared to the black line (estimated rate-distortion curve), respectively. We can observe that model (18) matches well the ground-truth distortion values.

Next, in Section 5 we further validate the effectiveness of the R-D model by applying it to the design of a modeling framework for compressive multi-path wireless video streaming, where a power-efficient problem is presented as an example.

5. NETWORK MODELING FRAMEWORK

We consider compressive wireless video streaming over multi-path multi-hop wireless multimedia sensor networks (WMSNs). Based on the R-D model developed in Section 4, we first formulate a video-quality-assured power minimization problem, and then solve the resulting nonlinear nonconvex optimization problem by proposing an online solution algorithm with low computational complexity.

Network model. In the considered WMSN there are a set $\mathcal{V}$ of camera sensors at the transmitter side, with each camera capturing a video sequence of the same scene of interest, and then sending the sequence to the server side through a set $\mathcal{Z}$ of pre-established multi-hop paths. Denote $\mathcal{L}^z$ as the set of hops belonging to path $z \in \mathcal{Z}$, with $d^{z,l}$ being the hop distance of the l^{th} hop in $\mathcal{L}^z$. Let $V = |\mathcal{V}|$, $Z = |\mathcal{Z}|$, and $L^z = |\mathcal{L}^z|$ represent cardinality of sets $\mathcal{V}$, $\mathcal{Z}$ and $\mathcal{L}^z$, respectively. The following three assumptions are considered:

- *Pre-established routing*, i.e., the set of multi-hop paths $\mathcal{Z}$ is established in advance through a given routing protocol (e.g., AODV [25]) and does not change during the video streaming session.
- *Orthogonal channel access*, i.e., there exists a pre-established orthogonal channel access, e.g., based on TDMA, FDMA, or CDMA, and hence concurrent transmissions do not interfere with each other [26].
- *Time division duplexing*, i.e., each node cannot transmit and receive simultaneously, implying that only half of the total air-time is used for transmission or reception.

At the receiver side, the video server concurrently and independently decodes each view of the received video sequences, and based on the reconstructed video sequences it then computes the rate control information and sends the information back to camera sensors for actual rate control. For this purpose, we define two types of video frames, Reference Frame (referred to as *R-frame*) and Non-Reference Frame (referred to as *NR-frame*). An R-frame is periodically transmitted by the R-view; all other frames sent out by the R-view and all frames transmitted by the NR-views are categorized as NR-frames. Compared to an NR-frame, an R-frame is encoded with equal or higher sampling rate and then sent to the receiver side with much lower transmission delay. Hence, an R-frame can be reconstructed with equal or higher video quality and used to estimate sparsity pattern information, which is then fed back to video cameras for rate control in encoding the following NR-frames. For the R-view, we consider a periodic frame pattern, meaning that the R-view camera encodes its captured video frames as R-frames periodically, e.g., one every 30 consecutive frames.

In the above setting, our objective is to minimize the average power consumption of all cameras and communication sensors in the network with guaranteed reconstructed video quality for each view, by jointly controlling video encoding rate and allocating the rate among candidate paths. To formalize this minimization problem, next we first derive the packet loss probability p_{loss} in (18).

Packet loss probability. According to the proposed modified R-D model (18), packet losses affect the video reconstruction quality because they introduce an effective measurement rate reduction. Therefore, effective estimation of packet loss probability at the receiver side has significant impact on frame-level measurement rate control.

In real-time wireless video streaming systems, a video packet can be lost primarily for two reasons: i) the packet fails to pass a parity check due to transmission errors introduced by unreliable wireless links, and ii) it takes too long for the packet to arrive at the receiver side, hence violating the maximum playout delay constraint. Denoting the corresponding packet loss probability as p_{per} and p_{dly}, respectively, the total packet loss rate p_{loss} can then be written as

$$p_{\text{loss}} = p_{\text{per}} + p_{\text{dly}}. \tag{19}$$

In the case of multi-path routing as considered above, p_{per} and p_{dly} in (19) can be further expressed as

$$p_{\text{per}} = \sum_{z \in \mathcal{Z}} \frac{b^z}{b} p_{\text{per}}^z, \tag{20}$$

$$p_{\text{dly}} = \sum_{z \in \mathcal{Z}} \frac{b^z}{b} p_{\text{dly}}^z, \tag{21}$$

where p_{per}^z and p_{dly}^z represent the packet loss rate for path $z \in \mathcal{Z}$ due to transmission error and delay constraint violation, respectively; b and b^z represent total video rate and the rate allocated to path $z \in \mathcal{Z}$, respectively.

Since each path $z \in \mathcal{Z}$ may have one or multiple hops, to derive the expressions for p_{per}^z and p_{dly}^z in (20) and (21), we need to derive the resulting packet error rate and delay violation probability at each hop l of path $z \in \mathcal{Z}$, denoted as $p_{\text{per}}^{z,l}$ and $p_{\text{dly}}^{z,l}$, respectively. For this purpose, we first express the feasible transmission rate achievable at each hop. For each hop $l \in \mathcal{L}^z$ along path $z \in \mathcal{Z}$, let $G^{z,l}$ and $N^{z,l}$ represent the channel gain that accounts for both path loss and fading, and the additive white Gaussian noise (AWGN) power currently measured by hop l, respectively. Denoting

$P^{z,l}$ as the transmission power of the sender of hop l, then the attainable transmission rate for the hop, denoted by $C^{z,l}(P^{z,l})$, can be expressed as [27]

$$C^{z,l}(P^{z,l}) = \frac{W}{2}\log_2\left(1 + K\frac{P^{z,l}G^{z,l}}{N^{z,l}}\right), \quad (22)$$

where W is channel bandwidth in Hz, calibration factor K is defined as

$$K = \frac{-\phi_1}{\log(\phi_2 p_{\text{ber}})}, \quad (23)$$

with ϕ_1, ϕ_2 being constants depending on available set of channel coding and modulation schemes, and p_{ber} is the predefined maximum residual bit error rate (BER). Then, if path $z \in \mathcal{Z}$ is allocated video rate b^z, for each hop $l \in \mathcal{L}^z$, the average attainable transmission rate should be equal to or higher than b^z, i.e.,

$$\text{E}[C^{z,l}(P^{z,l})] \geq b^z, \quad (24)$$

with $\text{E}[C^{z,l}(P^{z,l})]$ defined by averaging $C^{z,l}(P^{z,l})$ over all possible channel gains $G^{z,l}$ in (22).

Based on the above setting, we can now express the single hop packet error rate $p^{z,l}_{\text{per}}$ for each hop $l \in \mathcal{L}^z$ of path $z \in \mathcal{Z}$ as,

$$p^{z,l}_{\text{per}} = 1 - (1 - p_{\text{ber}})^L, \quad (25)$$

where L is the predefined packet length in bits. Further, we characterize the queueing behavior at each wireless hop as in [28] using a M/M/1 model to capture the effects of channel-state-dependent transmission rate (22) single-hop queueing delay. Denoting $T^{z,l}$ as the delay budget tolerable at each hop $l \in \mathcal{L}^z$ of path $z \in \mathcal{Z}$, the resulting packet drop rate due to delay constraint violation can then be given as [29]

$$p^{z,l}_{\text{dly}} = e^{-(\text{E}[C^{z,l}(P^{z,l})]-b^z)\frac{T^{z,l}}{L}}, \quad (26)$$

with $\text{E}[C^{z,l}(P^{z,l})]$ defined in (24). For each path $z \in \mathcal{Z}$, the maximum tolerable end-to-end delay $T^{\max}$ can be assigned to each hop in different ways, e.g., equal assignment or distance-proportional assignment [30]. We adopt the same delay budget assignment scheme as in [30].

Finally, given $p^{z,l}_{\text{per}}$ and $p^{z,l}_{\text{dly}}$ in (25) and (26), we can express the end-to-end packet error rate p^z_{per} and delay violation probability p^z_{dly} in (20) and (21) as, for each path $z \in \mathcal{Z}$,

$$p^z_{\text{per}} = \sum_{l\in\mathcal{L}^z} p^{z,l}_{\text{per}}, \ \forall z \in \mathcal{Z}, \quad (27)$$

$$p^z_{\text{dly}} = \sum_{l\in\mathcal{L}^z} p^{z,l}_{\text{dly}}, \ \forall z \in \mathcal{Z}, \quad (28)$$

by neglecting the second and higher order product of $p^{z,l}_{\text{per}}$ and of $p^{z,l}_{\text{dly}}$. The resulting p^z_{per} and p^z_{dly} provide an upper bound on the real end-to-end packet error rate and delay constraint violation probability. The approximation error is negligible if packet loss rate at each wireless hop is low or moderate. Note that it is also possible to derive a lower bound on the end-to-end packet loss rate, e.g., by applying the Chernoff Bound [31].

Packet loss to measurement rate. After having modeled p_{loss}, we now concentrate on determining κ to convert p_{loss} to measurement rate reduction (referred to as $R_d = \kappa \cdot p_{\text{loss}}$). First, parameter $\tau = \frac{1}{QN}$ is defined to convert the amount of transmitted bits of each frame to its measurement rate R used in the (18), with Q being the bit-depth for each measurement. We assume that b is equally distributed among F frames within 1 second for all V views, i.e., the transmitted bits for each frame is $b/F/V$. Thus, measurement rate R for each frame of each view is equal and defined as $R = \tau b/F/V$. Then, we can define κ as

$$\kappa = \tau L\lceil\frac{b/F/V}{L}\rceil, \quad (29)$$

and rewrite (18) as

$$D_{\text{dec}} = D_0 - \frac{\theta}{\tau b/F/V - \kappa p_{\text{loss}} - R_0}. \quad (30)$$

Problem formulation. Based on (30), we formulate, as an example of applicability of the proposed framework, the problem of power consumption minimization for quality-assured compressive multi-view video streaming over multi-hop wireless sensor networks, by jointly determining the optimal frame-level encoding rate and allocating transmission rate among multiple paths, i.e.,

$$\text{P}_4: \underset{P^{z,l},b^z,l\in\mathcal{L}^z,\forall z\in\mathcal{Z}}{\text{Minimize}} \sum_{z\in\mathcal{Z}}\sum_{l\in\mathcal{L}^z} P^{z,l} \quad (31)$$

$$\text{Subject to:} \quad b = \sum_{z\in\mathcal{Z}} b^z \quad (32)$$

$$D_{\text{dec}} \leq D_{\text{t}} \quad (33)$$

$$0 < \tau b/F/V - \kappa p_{\text{loss}} \leq 1 \quad (34)$$

$$0 \leq P^{z,l} \leq P_{\max}, \ \forall l \in \mathcal{L}^z, z \in \mathcal{Z}, \quad (35)$$

where D_{t} and $P_{\max}$ represent the constraints upon distortion and power consumption, respectively. Here, (33) and (34) are the constraints for required video quality level and total measurement rate not lower than 0 and higher than 1, respectively. In fact, the optimization problem P_4 is non-convex because the distortion constraint is non-convex. In Section 6, we adopt a solution algorithm to solve problem P_4 and demonstrate the reduction of the power saving by applying it based on the modeling framework.

6. PERFORMANCE EVALUATION

The topology includes a certain number V camera sensors and pre-established paths with random number of hops between camera sensors and the receiver. The frame rate is $F = 30$ fps, and the R-view periodically sends the R-frame every second. At the sparsity-aware CS independent encoder side, each frame is partitioned into 16×16 non-overlapped blocks implying $N_d = 256$. A measurement matrix $\mathbf{\Phi}^i_{vf}$ with elements drawn from independent and identically distributed (i.i.d) Gaussian random variables is considered, where the random seed is fixed for all experiments to make sure that $\mathbf{\Phi}^i_{vf}$ is drawn from the same matrix. The elements of the measurement vector $\tilde{\mathbf{y}}^i_{vf}$ are quantized individually by an 8-bit uniform scalar quantizer and then transmitted to the decoder. At the independent decoder end, we use $\mathbf{\Psi}_b$ composed of DCT transform basis as sparsifying matrix and choose the LASSO algorithm for reconstruction motivated by its low-complexity and excellent recovery performance characteristics. We consider two test multi-view sequences, *Exit* and *Vassar*, which are made publicly available [32]. In the sequences considered, the optical axis of each camera is parallel to the ground, and each camera is 19.5 cm away from its left and right neighbors. A spatial resolution of $(H = 240) \times (W = 320)$ is considered. *Exit* and *Vassar* are indoor surveillance and outdoor surveillance videos, respectively. The texture change of *Exit* is faster than that of *Vassar*, i.e., the block sparsity of *Exit* changes more quickly.

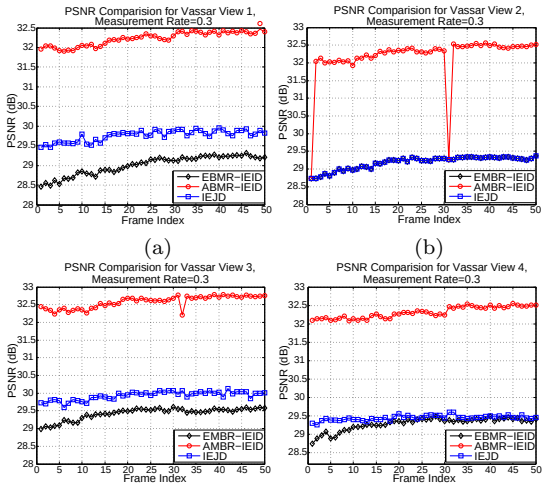


Figure 5: PSNR against frame index for (a) view 1, (b) view 2 (R-view), (c) view 3, and (d) view 4 of sequence *Vassar*.

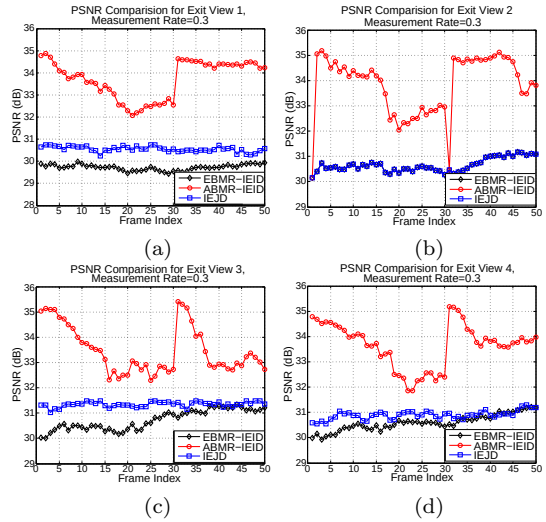


Figure 6: PSNR against frame index for (a) view 1, (b) view 2 (R-view), (c) view 3, and (d) view 4 of sequence *Exit*.

6.1 Evaluation of CS-based Multi-view Encoding/Decoding Architecture

We first experimentally study the performance of the proposed CS-based multi-view encoding/decoding architecture by evaluating the PSNR (as well as SSIM) of the reconstructed video sequences. Experiments are carried out only on the luminance component. Next, we illustrate the performance comparisons among (i) traditional Equal-Block-Measurement-Rate Independently Encoding and Independently Decoding approach (referred to as EBMR-IEID), (ii) the proposed sparsity-aware Adaptive-Block-Measurement-Rate Independently Encoding and Independently Decoding approach (referred to as ABMR-IEID) and (iii) Independently Encoding and Jointly Decoding (referred to as IEJD) proposed in [15] which selects one view as reference view reconstructed by traditional CS recovery method, while other views are jointly reconstructed by using reference frame.

Figures 5 and 6 show the PSNR comparisons of 50 frames for views 1, 2, 3 and 4 of *Vassar* and *Exit* multi-view sequences, where a 0.3 measurement rate for each view of ABMR-IEID and EBMR-IEID is selected. To assure fair comparison, the measurement rate of each view in IEJD is also set to 0.3. Besides, according to the R-view selection algorithm, view 2 is chosen as the R-view for this scenario. Since the R-view transmits the R-frame periodically, i.e., per second, and for the first frame of each period, the encoder will not encode them based on sparsity pattern, therefore we can observe drops occurred periodically in Fig. 5(b) and Fig. 6(b). For the *Vassar* sequences, as illustrated in Fig. 5, we can see that the proposed method ABMR-IEID outperforms the traditional approach EBMR-IEID and IEJD by up to 3.5 dB and 2.5 dB in terms of PSNR,respectively. For the *Exit* sequences, Figure 6 shows improvement in the reconstruction quality of ABMR-IEID compared with EBMR-IEID and IEJD fluctuates more than that of *Vassar* video, with increased PSNR varying from 5 dB to 2 dB and from 4 dB to 1 dB, respectively. This phenomenon occurs because of the video-based features, i.e., the texture of *Exit* changes faster than in *Vassar*. In other words, the proposed scheme is more robust in surveillance scenarios where the changes of texture are less severe. However, we can eliminate this phenomenon by transmitting R-frames more frequently. Figures 5 and 6 also depict performance improvement on NR-views (views 1, 3 and 4 here), i.e., by sharing the sparsity information between R-view and NR-views, correlation among views is implicitly exploited to improve the reconstruction quality.

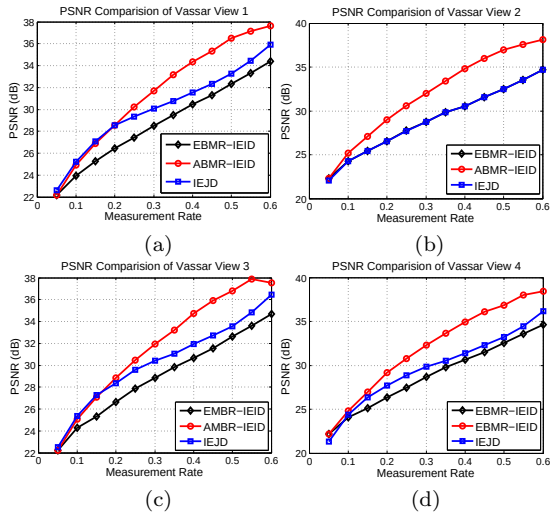
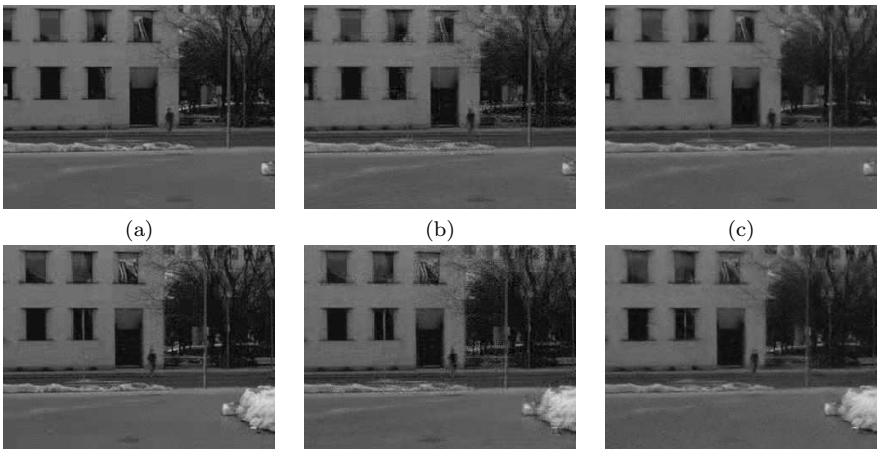


Figure 7: Rate-distortion comparison for frame 75 of *Vassar* sequences: (a) view 1, (b) view 2, (c) view 3, and (d) view 4.

We then illustrate the rate-distortion characteristics of ABMR-IEID, EBMR-IEID and IEJD. Figure 7 shows the comparisons of 4-view scenario, where the 75th frame of *Vassar* is taken as example. Evidently, ABMR-IEID outperforms significantly EBMR-IEID and IEJD, especially as the number of measurements increases. We can observe that at measurement rate 0.4, ABMR-IEID can improve PSNR by up to 4.4 dB and 2.4 dB, not only on R-view but also on NR-views. In the experiments, as the number of views increases, ABMR-IEID can still obtain significant PSNR gain compared to EBMR-IEID; while the performance of IEJD degrades faster as the distance to R-view increases, which can be apparently observed from Figs. 5, 6 and 7 where view 4 has distance 2 to R-view but with relatively lower PSNR gain compared to views 1 and 3.

Next, we extend the scenario to 8 views on *Vassar*, where view 4 is selected as R-view, and the measurement rate is set to 0.35 for all views. Figure 8 shows the specific recon-



(a) (b) (c) (d) (e) (f)

Figure 8: Reconstructed frame 25 of view 3 by (a) ABMR-IEID, (b) EBMR-IEID, (c) IEJD, and reconstructed frame 25 of view 7 by (d) ABMR-IEID, (e) EBMR-IEID, and (f) IEJD.

Table 3: PSNR and SSIM comparison for *Vassar* eight views.

View #	ABMR-IEID		EBMR-IEID		IEJD	
	PSNR (dB)	SSIM	PSNR (dB)	SSIM	PSNR (dB)	SSIM
1	33.6675	0.8648	30.0883	0.8215	30.2717	0.7887
2	33.7768	0.8686	30.3459	0.8262	30.3355	0.7902
3	34.1934	0.8771	30.6265	0.8323	30.9214	0.8106
4	33.5766	0.8696	30.4168	0.8294	30.4168	0.8294
5	33.3030	0.8624	30.1011	0.8169	30.3641	0.7909
6	34.2191	0.8846	30.6803	0.8382	30.7265	0.8059
7	32.9924	0.8575	29.8250	0.8162	29.6648	0.7772
8	32.3376	0.8472	29.3713	0.8054	29.5466	0.7742

structed image comparison, where the left column illustrates the reconstructed frame 25 of view 3 and view 7 by ABMR-IEID, respectively. The milldle column shows the reconstructed images by EBMR-IEID, and the left columns shows the results obtained by using IEJD. We can observe that the quality of images located in the left column is much better than that in the right two columns (e.g., the curtain in the 2nd floor and person in the scene, and etc.). Furthermore, Table 3 shows the detailed PSNR and SSIM value comparison between ABMR-IEID and EBMR-IEID and IEJD for frame 25 of 8 views. From Fig. 8 and Table 3, we can see that ABMR-IEID also works well on 8 views compared to ABMR-IEID and EBMR-IEID, with PSNR and SSIM improvement up to 3.5 dB and 0.05, respectively. However, the IEJD method proposed in [15] does not perform well on 8 views, where the gain is almost negligible.

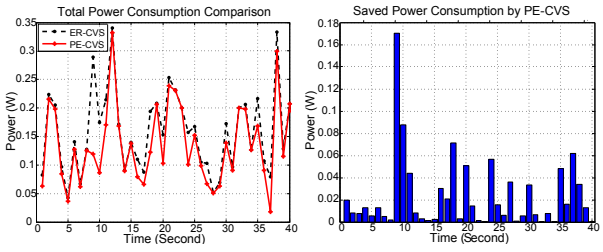
6.2 Evaluation of Power-efficient Compressive Video Streaming

The following network topology is considered: 2-path scenario with 2-hop path 1 and 1-hop path 2. We assume bandwidth $W = 1$ MHz for each channel. The maximum transmission power at each node is set to 1 W and the target distortion in MSE is 50. We also assume the maximum end-to-end delay is $T^{\max} = 0.5$s assigned to each hop proportional to the hop distance. To evaluate PE-CVS (referred to as the proposed power-efficient compressive video streaming algorithm), we compare it with an algorithm (referred to as ER-CVS) that equally splits the frame-level rate calculated by PE-CVS onto different paths.

Figure 9 illustrates the total power consumption comparison between PE-CVS and ER-CVS and the saved power by PE-CVS compared to ER-CVS. From Fig. 9(a), we see that PE-CVS (depicted in red line) results in less power consumption than ER-CVS (black dash line). At some points, the total power consumption of PE-CVS and ER-CVS is almost the same. This occurs because the path-level bit rates calculated by PE-CVS are equal to each other. Since ER-CVS uses frame-level rate obtained from PE-CVS and equally allocates it to each path, thereby resulting in the same power consumption. As shown in Fig. 9(b), the histogram apparently shows that PE-CVS saves more power than ER-CVS, up to 170 mW.

7. CONCLUSIONS

We have proposed a novel compressive multi-view video coding/decoding architecture - cooperative sparsity-aware independent encoder and independent decoder. We also introduced a central controller to do the sparsity pattern estimation, R-view selection, mean value estimation and implement network optimization algorithms. By introducing limited channel feedback and enabling lightweight sparsity information sharing between R-view and NR-views, the encoders independently encode the video sequences with sparsity awareness and exploit multi-view correlation to improve the reconstruction quality of NR-views. Based on the proposed encoding/decoding architecture, we developed an R-D model that considers the packet loss effect in C-



(a) (b)

Figure 9: 2-path Scenario: (a) Total power consumption comparison, (b) Saved power consumption by PE-CVS compared to ER-CVS.

S video streaming in WSNs. Then, we studied a modeling framework to design network optimization algorithms, where packet loss rate for a multi-hop multi-path sensor network and the conversion from packet loss rate to the measurement rate reduction are derived. Finally, we presented a power-efficient algorithm. Extensive simulation results showed that the designed compressive multi-view framework can considerably improve the video reconstruction quality with minimal power consumption.

8. ACKNOWLEDGMENTS

This paper is based upon work supported in part by the US National Science Foundation under grants CNS1117121 and CNS1218717 and by the Office of Naval Research under grant N00014-11-1-0848. Zhangyu Guan was also supported in part by the Shandong Doctoral Foundation under grant 2012BSE27052.

9. REFERENCES

[1] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury. A Survey on Wireless Multimedia Sensor Networks. *Computer Networks*, 51(4):921–960, March 2007.

[2] Z. Guan and T. Melodia. Cloud-Assisted Smart Camera Networks for Energy-Efficient 3D Video Streaming. *IEEE Computer*, 47(5):60–66, May 2014.

[3] M. M. Hannuksela, D. Rusanovskyy, W. Su, L. Chen, R. Li, P. Aflaki, D. Lan, M. Joachimiak, H. Li, and M. Gabbouj. Multiview-Video-Plus-Depth Coding Based on the Advanced Video Coding Standard. *IEEE Transactions on Image Processing*, 22(9):3449–3458, September, 2013.

[4] A. Vetro, T. Wiegand, and G. J. Sullivan. Overview of the Stereo and Multiview Video Coding Extensions of the H.264/MPEG-4 AVC Standard. *Proceedings of the IEEE*, 99(4):626–642, April 2011.

[5] S. Pudlewski, N. Cen, Z. Guan, and T. Melodia. Video Transmission over Lossy Wireless Networks: A Cross-layer Perspective. *IEEE Journal of Selected Topics in Signal Processing*, 9(1):6–22, February 2015.

[6] E. J. Candes and M. B. Wakin. An Introduction to Compressive Sampling. *IEEE Signal Processing Magazine*, 25(2):21–30, March 2008.

[7] D. L. Donoho. Compressed Sensing. *IEEE Trans. on Information Theory*, 52(4):1289–1306, April 2006.

[8] S. Pudlewski and T. Melodia. A Tutorial on Encoding and Wireless Transmission of Compressively Sampled Videos. *IEEE Communications Surveys & Tutorials*, 15(2):754–767, Second Quarter 2013.

[9] H. Chen, L. Kang, and C. Lu. Dynamic measurement rate allocation for distributed compressive video sensing. In *Proc. IEEE/SPIE Visual Communications and Image Processing (VCIP)*, Huangshan, China, July 2010.

[10] Y. Liu, M. Li, and D. A. Pados. Motion-aware Decoding of Compressed-sensed Video. *IEEE Trans. on Circuits System Video Technology*, 23(3):438–444, March 2013.

[11] S. Pudlewski, T. Melodia, and A. Prasanna. Compressed-Sensing-Enabled Video Streaming for Wireless Multimedia Sensor Networks. *IEEE Trans. on Mobile Computing*, 11(6):1060–1072, June 2012.

[12] S. Pudlewski and T. Melodia. Compressive Video Streaming: Design and Rate-Energy-Distortion Analysis. *IEEE Transactions on Multimedia*, 15(8):2072–2086, December 2013.

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

[14] M. Trocan, T. Maugey, E. W. Tramel, J. E. Fowler, and B. Pesquet-Popescu. Compressed Sensing of Multiview Images Using Disparity Compensation. In *Proc. Intern. Conf. on Image Processing (ICIP)*, Hong Kong, September 2010.

[15] N. Cen, Z. Guan, and T. Melodia. Joint Decoding of Independently Encoded Compressive Multi-view Video Streams. In *Proc. of Picture Coding Symposium (PCS)*, San Jose, CA, December 2013.

[16] C. Li, D. Wu, and H. Xiong. Delay-Power-Rate-Distortion Model for Wireless Video Communication Under Delay and Energy Constraints. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(7):1170–1183, July 2014.

[17] Z. He, Y. Liang, L. Chen, I. Ahmad, and D. Wu. Power-rate-distortion Analysis for Wireless Video Communication under Energy Constraints. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(5):645–658, May 2005.

[18] 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*, 1(4):586–598, December 2007.

[19] R. Tibshirani. Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1996.

[20] D. L. Donoho, M. Elad, and V. N. Temlyakov. Stable Recovery of Sparse Overcomplete Representations in the Presence of Noise. *IEEE Transactions on Information Theory*, 52(1):6–18, January 2006.

[21] K. Stuhlmuller, N. Farber, M. Link, and B. Girod. Analysis of Video Transmission over Lossy Channels. *IEEE Journal on Selected Areas in Communications*, 18(6):1012–1032, June 2000.

[22] D. Slepian and J. K. Wolf. Noiseless Coding of Correlated Information Sources. *IEEE Transactions on Information Theory*, 19(4):471–480, July 1973.

[23] A. D. Wyner and J. Ziv. The Rate-distortion Function for Source Coding with Side-information at the Decoder. *IEEE Transactions on Information Theory*, 22(1):1–10, January 1976.

[24] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004.

[25] Charles E. Perkins and Elizabeth M. Royer. Ad hoc On-Demand Distance Vector (AODV) Routing. *RFC 3561*, July 2003.

[26] Y. Shi, S. Sharma, Y. T. Hou, and S. Kompella. Optimal Relay Assignment for Cooperative Communications. In *Proc. ACM Intern. Symp. on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Hong Kong, China, May 2008.

[27] A. Goldsmith. *Wireless Communications*. Cambridge University Press, New York, NY, USA, 2005.

[28] X. Zhu, E. Setton, and B. Girod. Congestion-distortion Optimized Video Transmission over Ad Hoc Networks. *EURASIP Journal of Signal Processing: Image Communications*, 20:773–783, 2005.

[29] D. Bertsekas and R. Gallager. *Data Networks*. Prentice Hall, USA, 2000.

[30] T. Melodia and I. D. Akyildiz. Cross-layer Quality of Service Support for UWB Wireless Multimedia Sensor Networks. In *Proc. of IEEE Conference on Computer Communications (INFOCOM)*, Phoenix, AZ, April 2008.

[31] H. Chernoff. A Measure of Asymptotic Efficiency for Tests of a Hypothesis Based on The Sum of Observations. *Ann. Math. Statist.*, 23(4):493–507, December 1952.

[32] Mitsubishi Electric Research Laboratories. MERL Multi-view Video Sequences. [Available] ftp://ftp.merl.com/pub/avetro/mvc-testseq.