# Compressive Video Streaming: Design and Rate-Energy-Distortion Analysis

Scott Pudlewski and Tommaso Melodia

***Abstract*—Real-time encoding and error-resilient wireless transmission of multimedia content using traditional encoding techniques requires relatively high processing and transmission power, while pervasive surveillance and monitoring systems often referred to as wireless multimedia sensor networks (WMSNs) [1] are generally composed of low-power, low-complexity devices. To bridge this gap, this article introduces and analyzes a compressive video sensing (CVS) encoder designed to reduce the required energy and computational complexity at the source node. The proposed encoder leverages the properties of compressed sensing (CS) to overcome many of the limitations of traditional encoding techniques, specifically lack of resilience to channel errors, and high computational complexity. Recognizing the inadequacy of traditional rate-distortion analysis to account for the constraints introduced by resource-limited devices, we introduce the notion of rate-energy-distortion, based on which we develop an analytical/empirical model that predicts the received video quality when the overall energy available for both encoding and transmission of each frame of a video is fixed and limited and the transmissions are affected by channel errors. The model allows comparing the received video quality, computation time, and energy consumption per frame of different wireless streaming systems, and can be used to determine the optimal allocation of encoded video rate and channel encoding rate for a given available energy budget. Based on the proposed model, we show that the CVS video encoder outperforms (in an energy constrained system) two common encoders suitable for a wireless multimedia sensor network environment; H.264/AVC intra and motion JPEG (MJPEG). Extensive results show that CVS is able to deliver video at good quality (an SSIM value of 0.8) through lossy wireless networks with lower energy consumption per frame than competing encoders.**

***Index Terms*—Compressed sensing, video surveillance, video encoding, multimedia sensor networks.**

## I. Introduction

RECENT advances in sensing, computation, storage, and wireless networking are driving an increasing interest in multimedia [1], [3] and people-centric [4], [5] sensing applications. Wireless Multimedia Sensor Networks (WMSN) are self-organizing systems of embedded devices deployed to retrieve, distributively process in real-time, store, correlate, and fuse multimedia streams originated from heterogeneous sources [6]. WMSNs are enablers for applications including video surveillance, storage and subsequent retrieval of potentially relevant activities.

While applications of multimedia and participatory sensor networks show high promise, they require wirelessly networked streaming of video originating from devices that are constrained in terms of instantaneous power, energy storage, memory, and computational capabilities. While there has been intense research and considerable progress in solving numerous wireless sensor networking challenges, the underlying root problem of enabling real-time quality-aware video streaming in large-scale, possibly multi-hop, wireless networks of embedded devices is still substantially open. State-of-the-art technology, for the most part based on streaming predictively encoded video (e.g., MPEG-4 Part 2, H.264/AVC [7]–[9], H.264/SVC [10]) through a layered wireless communication protocol stack, is affected by the following fundamental limitations:

- **Predictively Encoded Video is not Resilient to Channel Errors**. In existing layered protocol stacks (e.g., based on the IEEE 802.11 and 802.15.4 standards), frames are split into multiple packets. If even a single bit is flipped due to channel errors, after a cyclic redundancy check, the entire packet is dropped at a final or intermediate receiver.[1] This can lead to the video decoder being unable to decode an independently coded $(I)$ frame, thus leading to the loss of an entire sequence of video frames. Structure in video representation, which plays a fundamental role in our ability to compress video, is detrimental when it comes to wireless video transmission with lossy links. Ideally, when one single bit is in error, we would like the effect on the reconstructed video to be *unperceivable*. In addition, the perceived video quality should *gracefully and proportionally degrade* with decreasing channel quality.
- **High Power Consumption and Encoder Complexity on Embedded Devices**. State-of-the-art predictive encoding requires finding motion vectors, which is a computationally intensive operation at the encoder. This naturally leads to high energy consumption at the encoder, and/or high processor load or additional costs for specialized processors [1]. *New video encoding paradigms are needed to reverse the traditional balance of complex encoder and simple decoder, which is fundamentally unsuited for embedded video sensing*.

Manuscript received December 11, 2011; revised September 06, 2012 and December 12, 2012; accepted December 14, 2012. Date of publication August 30, 2013; date of current version November 13, 2013. A preliminary shorter version of this paper [2] appeared in the Proceedings of IEEE GLOBECOM 2011, Houston, TX, USA, December 2011. This paper is based upon work supported in part by the National Science Foundation under grant CNS1117121 and by the Office of Naval Research under grant N00014-11-1-0848. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Eckehard G. Steinbach.

The authors are with the Department of Electrical Engineering, State University of New York (SUNY) at Buffalo, Buffalo, NY 14260 USA (e-mail: smp25@buffalo.edu; tmelodia@buffalo.edu).

Digital Object Identifier 10.1109/TMM.2013.2280245

[1]No forward error correction (FEC) is used in either IEEE 802.11 or 802.15.4, and hence a faulty bit may corrupt the entire packet. In cellular transmissions, data can be protected in case of unicast transmissions, but in case of video multicast with multiple receivers that differ in link quality, it is not feasible to select a unique channel coding rate that fits all receivers. For the performance evaluation section if this paper, we do use an RCPC FEC code at the application layer.

To deal with these limitations, we design and propose the compressive video sensing (CVS) video encoder, which is based on the theory of compressed sensing (CS) [11]–[14]. Compressed sensing (aka "compressive sampling") is a new paradigm that allows the faithful recovery of signals from $M \ll N$ measurements where N is the number of samples required for the Nyquist sampling. Hence, CS can offer an alternative to traditional video encoders by enabling imaging systems that sense and compress data simultaneously *at very low computational complexity for the encoder*. CS images and video are also resilient to bit errors [15]. Based on the low-complexity and high error resilience of CS signals, CVS is designed to compress video with low energy consumption and computational complexity at the video source.

Typically, video encoders are evaluated in terms of rate-distortion performance. However, there are some significant limitations to rate-distortion analysis when applied to video encoding on resource limited systems—above all, computational complexity and energy consumption are not considered in traditional rate-distortion analysis. However, these factors can play a key role in both the suitability of an encoder for a specific application as well as the ability of that encoder to be implemented. Energy consumption, *including energy consumption required for encoding*, needs to be considered jointly with the rate-distortion to obtain a valid and realistic assessment of whether an encoder is suited for implementation in a WMSN. We show that error resilience, along with the decreased computational complexity, allow CVS to perform very well in terms of *energy-rate-distortion* performance.

To evaluate this, we conduct an experiment-driven analysis of the energy-rate-distortion performance of CVS, along with two traditional video encoders designed for embedded wirelessly networked devices. Different from previous work on low-complexity encoding [16], [17], we jointly consider the effects of processing on resource-constrained devices and of wireless transmission on the performance of wireless encoders. We first develop an analytical model that can be manipulated to determine, for a given total energy budget per frame and a given channel condition, the optimal joint allocation of energy between wireless transmission and video encoding. Intuitively, for a fixed energy budget, as more energy is allocated to the encoder (resulting in less compression and a video of better quality), less energy is available to transmit that video over a wireless link, which will potentially result in an increased bit error rate and lower quality at the receiver. Conversely, as more energy is allocated to transmission, less energy is available to encode the video, resulting in a lower quality video. While this work is based on modeling the tradeoff between computation energy and transmission energy, real system parameters are used to develop a model that can accurately predict how a real system will perform in a WMSN. The developed model is used to find, for a given encoder, the optimal allocation between these two components, along with the optimal channel coding rate, which results in the optimal received video quality.

A reference scenario is illustrated in Fig. 1. This diagram shows how the received video quality for a given video encoder is based on the specific encoding rate, the family of channel codes (and specific channel encoding rate), and the given energy budget per frame. The energy budget per frame is split between the energy needed to encode the video and the energy required for transmitting the video over the lossy channel. These choices affect the quality of the received video at the multimedia sink. We present a methodology to compare different encoders in terms of the energy budget required to obtain a target video quality at the receiver. We then present a model to find the optimal encoded video rate and the channel coding rate that result in the maximum received video quality for a given energy budget.

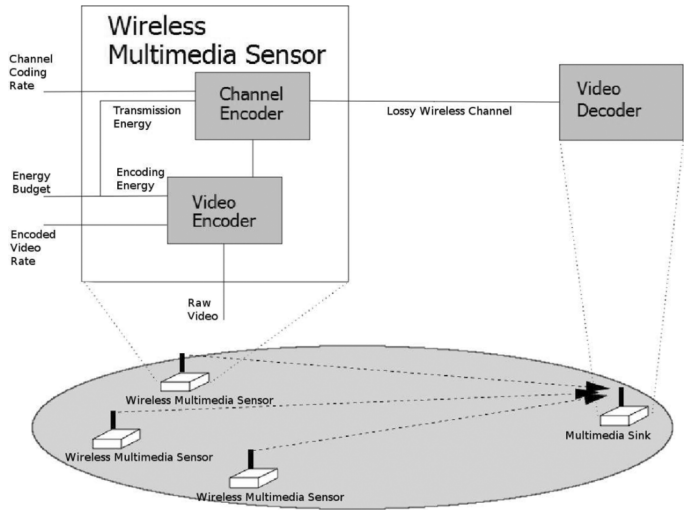


Fig. 1. Energy-Aware Video Encoding and Transmission.

For comparison with CVS, we focus on two video encoders with different characteristics mainly in terms of their complexity and of the resulting rate-distortion performance. The first is Motion JPEG (MJPEG), which is a simple low-complexity encoder designed for low-power or portable devices. For this paper we use the implementation in [18]. MJPEG is a video encoder in which each frame is individually encoded according to the JPEG standard. Because it does not exploit dependency between frames, motion estimation between frames is not necessary, resulting in lower encoding complexity. We also compare CVS to intra-encoded H.264/AVC, which is the standard H.264/AVC using intra-frame prediction only.

While standard inter-encoded H.264 would clearly have a much better rate-distortion performance, there are two important reasons for only considering the intra-encoded H.264/AVC. The most important reason is that the focus of a WMSN system is to transmit a "good" quality video (where "good" will be more clearly defined later) while extending the network lifetime (i.e., reducing the power consumption) as much as possible. As we will show later, the energy required to encode even intra-encoded H.264/AVC is the limiting factor in the performance of this codec in a WMSN. This intuition was validated in preliminary experiments based on those presented in Section VII. Because the measured encoding energy was more than a factor of 10 higher when motion vector calculations were included, for the scale presented there was *not enough energy budget available to encode any video at all*. This makes full H.264/AVC an unrealistic and impractical choice for encoding video in a

WMSN. Even if full H.264/AVC were able to be implemented, it would perform similar to intra-encoded H.264/AVC with better rate-distortion performance at the cost of more energy required for encoding. It would require more energy to implement the inter-frame prediction, it would be more sensitive to channel errors, but it would result in a smaller encoded size.

The remainder of this paper is structured as follows. In Section II, we discuss related work in video using compressed sensing. In Section III, we contain the imaging portion of the CVS video encoder, which is extended to video encoding is Section IV. In Section V we present the empirical models used to characterize the video encoders, and in Section VI we present the video quality model. Finally, the performance results of the three encoders are presented in Section VII, while in Section VIII we draw the main conclusions and discuss future work.

## II. Related Work

In this section, we will briefly discuss related work in video using compressed sensing. Currently, there are a number of different approaches to CS video encoding. We present a few notable encoders based either entirely or partially on CS.

**Distributed Compressive Video Sensing:** Recent work in distributed video coding (DVC) [19] has shown that error correction techniques can be used to transfer the majority of the complexity of video encoding to the receiver. This is then extended in [20] to CS encoded images. To accomplish this, the authors first assume that there are two sequential video frames $W$ and $S$. Regardless of the amount of motion in the video, there will usually be at least some correlation between $W$ and $S$. For many types of video common in WMSNs (such as security or surveillance videos), the correlation will be very high. This allows frame $S$ to be viewed as a *corrupted version of frame* $W$. In other words, $S$ can be viewed as a version of $W$ that has been transmitted through a lossy channel. Error correction bits can then be created and used to "correct" the differences between the two frames.

The authors of [20] then propose a modification of the stopping criteria of the gradient projection for sparse reconstruction (GPSR) algorithm so the quality of the reconstructed video frame is kept sufficiently high without incurring excessive computation. The proposed criterion minimizes the difference between the reconstructed frame and the side information generated at the receiver. This is different from standard GPSR, which generally terminates when the norm of the minimum between the reconstructed signal and the gradient of the reconstructed signal is below a tolerance. The authors show that using the proposed criteria results in up to 4 dB PSNR higher quality than standard GPSR. While this approach does show promise, the benefits are based on improvements to the CS reconstruction algorithm at the receiver. In contrast, CVS works at the encoder and is independent of the reconstruction algorithm. Any improvements to CS reconstruction techniques will also improve the quality of a CVS encoder.

**Block Based CS Video:** The single pixel camera was introduced in [21] for imaging applications. This imaging system is immediately applicable to video acquisition [22]. The key is that each measurement is taken *sequentially in time*, i.e., each CS sample represents a *specific moment* in time. Since a video is a 3D signal which is a sequence of 2D images, each measurement is a sample of an individual 2D image.

The authors of [22] present two methods for reconstructing the frames into a video. First, each "frame" is created by an aggregation process and reconstructed independently. While simple, this process essentially ignores any temporal correlation between frames. In addition, any fast motion between frames could cause severe problems in the reconstruction of the image. However, a second method is presented that does take advantage of temporal correlation. A 3D wavelet is used as the sparsifying transform and the entire "block" of video is reconstructed at once.

While such a scheme is very promising, there is one major limitation. The complexity of the reconstruction process is highly nonlinear (traditional interior point methods have a complexity of $O(M^2N^{3/2})$, where $N$ is the length of the raw frame, and $M$ is the length of the frame after compression). So while this scheme will clearly result in very good performance in terms of quality, reconstructing the video in real time is not practical with currently available hardware. In addition, because of this dependence on the single pixel camera, the image compression scheme is limited to the operations that can be performed at that camera. CVS could work with an extension on the single pixel camera, or can be implemented using a traditional CMOS camera. This allows CVS to implement simple operations (i.e., the vector subtraction necessary for the difference vector calculation) that can more directly take advantage of temporal correlation. However, if there is an application that does not require real time reconstruction, [22] presents a system that requires very low computational complexity at the sensor and will perform very well in terms of rate distortion performance.

**Hybrid CS Video Encoders:** Some recent work has attempted to combine CS concepts with traditional video encoding concepts. While these systems may not be appropriate for WMSN applications, some of them are still worth mentioning as they introduce innovative concepts that could be applied to future CS video systems more appropriate for WMSNs. The main limitations of these systems compared to entirely CS based systems is the complexity required to both encode and capture video.

Two examples of this are Distributed Compressed Video Sensing (DISCOS) [23] and the block-based CS encoder presented in [24]. DISCOS divides a video into two types of frames; key frames (or $I$-frames) and non-key frames (called $CS$-frames). The scheme uses standard video compression (such as MPEG/H.26x intra-encoding) on key frames. The non-key frames, however, are sampled with CS using a combination of both frame-based measurement (linear combinations of the entire frame) and block-based measurements (linear combinations of a set of pixels restricted to a set of non-overlapping blocks). The encoder presented in [24] also presents a block based compressive video sampling system. In this work, as in DISCOS above, a key frame is sampled and encoded using traditional methods. This key frame is then divided into non-overlapping blocks, which are each analyzed for "local sparsity". Only blocks determined to be sparse *enough* are

encoded using CS, while the rest are encoded using traditional methods. Non-key frames are then encoded using either CS or traditional methods based on the analysis of the key frame. Unlike CVS, these systems both incorporate CS concepts into the video encoding system. However, both require traditional video or image encoding methods as the basis for the compression.

**Compressed-Sensing-Enabled Video Streaming:** A preliminary version of this encoder was first introduced in [14]. Unlike this paper, the focus of [14] was on the development of a transport layer rate control policy that implicitly solved a sum utility optimization problem in a distributive manner. In this paper, the CVS encoder is presented in full detail and the energy-rate-distortion performance is compared to other encoders appropriate for WMSNs.

## III. Compressive Imaging

CVS first encodes each individual frame independently as an image. In this section we will first examine properties of images compressed using CVS, and discuss how they contribute to the energy-rate-distortion performance of CS based video encoding.

### A. Compressed Sensing Basics

We first introduce the basic concepts of compressed sensing as applied to images. We consider an image signal represented as $\mathbf{x} \in \mathbb{R}^N$, where $N$ is the number of pixels in the image and each element $x_i$ represents the $i^{th}$ pixel in the raster scan of the image. We assume that there exists an invertible transform matrix $\Psi \in \mathbb{R}^{N \times N}$ such that

$$\mathbf{x} = \Psi \mathbf{s} \tag{1}$$

where $\mathbf{s}$ is a $K$-sparse vector, i.e., $\|\mathbf{s}\|_0 = K$ with $K < N$, and where $\|\cdot\|_p$ represents $p$-norm. This means that the image has a sparse representation in some transformed domain, e.g., wavelet [25] or DCT [26]. The signal is measured by taking $M < N$ samples of the element vectors through a linear measurement operator $\Phi \in \mathbb{R}^{M \times N}$. The resulting sample vector $\mathbf{y}$ is then defined as

$$\mathbf{y} = \Phi \mathbf{x} = \Phi \Psi \mathbf{s} = \tilde{\Psi} \mathbf{s}. \tag{2}$$

We would like to recover $\mathbf{x}$ from measurements in $\mathbf{y}$. However, since $M < N$, the system is underdetermined. Given a solution $\mathbf{s^0}$ to (2), any vector $\mathbf{s}^*$ such that $\mathbf{s}^* = \mathbf{s}^0 + \mathbf{n}$, and $\mathbf{n} \in \mathcal{N}(\tilde{\Psi})$ (where $\mathcal{N}(\tilde{\Psi})$ represents the null space of $\tilde{\Psi}$), is also a solution to (2). However, it was proven in [12] that if the measurement matrix $\Phi$ is sufficiently incoherent with respect to the sparsifying matrix $\Psi$, and $K$ is smaller than a given threshold (i.e., the sparse representation $\mathbf{s}$ of the original signal $\mathbf{x}$ is "sparse enough"), then the original $\mathbf{s}$ can be recovered by solving

$$\begin{aligned} \underset{\mathbf{s}}{\text{minimize}} \quad & \|\mathbf{s}\|_0 \\ \text{subject to} \quad & \mathbf{y} = \tilde{\Psi}\mathbf{s}, \end{aligned} \tag{3}$$

i.e., by finding the sparsest solution that satisfies (2), i.e., the sparsest solution that "matches" the measurements in $\mathbf{y}$.

Unfortunately, finding the *sparsest* vector $\hat{\mathbf{s}}$ using (3) is in general NP-hard [27]. However, for matrices $\tilde{\Psi}$ with sufficiently incoherent columns, whenever this problem has a sufficiently sparse solution, the solution is unique, and it is equal to the solution of the following problem:

$$\begin{aligned} \underset{\mathbf{s}}{\text{minimize}} \quad & \|\mathbf{s}\|_1 \\ \text{subject to} \quad & \|\mathbf{y} - \tilde{\Psi}\mathbf{s}\|_2^2 < \epsilon, \end{aligned} \tag{4}$$

where $\epsilon$ is a small tolerance.

Formally, any sampling matrix $\Phi$ must satisfy the uniform uncertainty principle (UUP) [12], [28]. The UUP states that if $M$ is chosen such that

$$M \geq K \log N, \tag{5}$$

then for any $K$-sparse vector $\mathbf{s}$, the energy of the measurements $\Phi\mathbf{s}$ will be comparable to the energy of $\mathbf{s}$ itself:

$$\frac{1}{2}\frac{M}{N} \cdot \|\mathbf{s}\|_2^2 \leq \|\Phi\mathbf{s}\|_2^2 \leq \frac{3}{2}\frac{M}{N} \cdot \|\mathbf{s}\|_2^2. \tag{6}$$

To see the association between UUP and sparse reconstruction [28], suppose that (6) holds for sets of size 2K. Assume $\mathbf{s}_0$ is a $K$ sparse vector. There can not be any other $K$ sparse or sparser vector $\mathbf{s}' \neq \mathbf{s}_0$ that leads to the same measurements. If there were such a vector, then the difference $\mathbf{h} = \mathbf{s_0} - \mathbf{s}'$ would be $2K$-sparse and have $\Phi\mathbf{h} = 0$, which is not compatible with (6).

Note that (4) is a convex optimization problem [29]. The reconstruction complexity equals $O(M^2N^{3/2})$ if the problem is solved using interior point methods [30]. Although more efficient reconstruction techniques exist [31]–[36], we only discuss specific reconstruction algorithms when necessary to understand the specific imaging or video system. Otherwise, the discussions presented here are independent of the specific reconstruction algorithm.

### B. CVS Imaging

The section above describes how CS can be used to compress an image. In this section, we describe how the CS image can be used as the first component in a compressive video sensing (CVS) video encoder. This is done by examining the properties of CS encoded images when encoded in energy and complexity constrained systems and transmitted through lossy channels.

*1) Effects of Approximate Sparsity:* In Section III-A, we stated that any $K$-sparse signal sampled using (2) that satisfies (5) can be recovered using (4). However, wavelet (or DCT) transformed images are only *approximately* sparse. For example, Fig. 2 shows the DCT coefficients of the Lena image [37] sorted in increasing order. While the image is clearly compressible, few if any of the DCT coefficients are *exactly* zero.

When we use (4) to reconstruct Lena with $M < N$, the reconstruction process will force the smaller coefficients to be exactly zero [13], which will cause distortion in the reconstructed image. We can see how this affects the quality of the reconstructed image by measuring the effect of this sparse approximation on DCT transformed images. The results of this test are shown in Fig. 3. This figure was created by finding the DCT transform of the Lena image, forcing the smallest coefficients to zero and finding the inverse transform of the result. As more coefficients are forced to zero, the quality of the reconstructed image decreases.

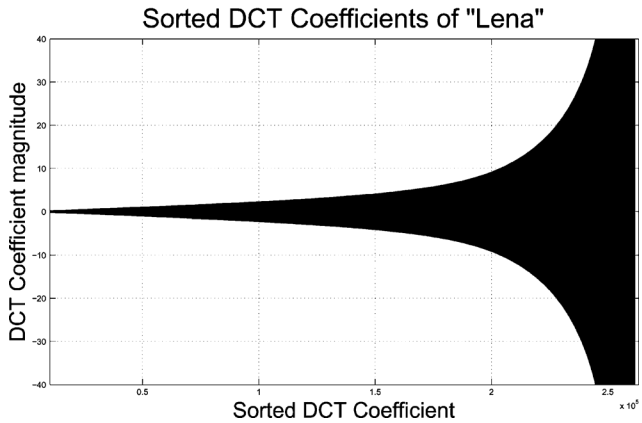


Fig. 2. DCT coefficients of Lena sorted in ascending order.

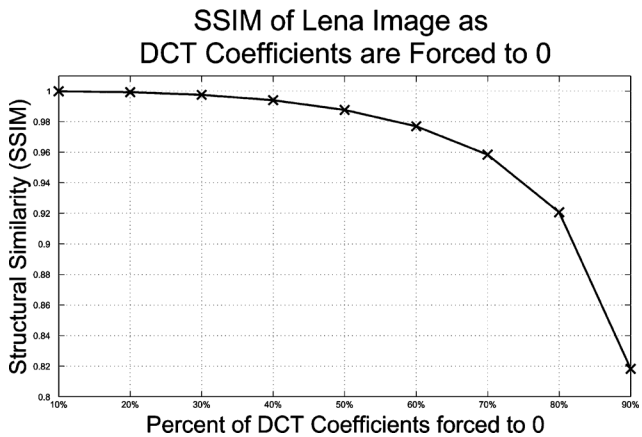


Fig. 3. SSIM of Lena after DCT transform, forcing the smallest coefficients to zero and inverse DCT transform.

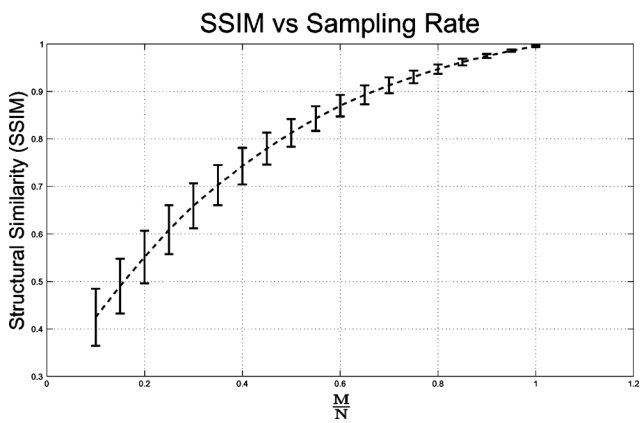


Fig. 4. SSIM vs. sampling rate $\frac{M}{N}$.

In practice, this means that, unlike the sparse case described above, "exact" recovery is not possible. Instead, as more samples are used in the reconstruction (i.e., as $M$ approaches $N$), the reconstructed image quality increases. This is demonstrated in Fig. 4, which shows the mean of the received quality over all of the images in the USC SIPI database [37] encoded using (2). These tests were done using the wavelet transform as the sparsifying transform and reconstructed using the GPSR [33] algorithm. As $M$ is increased and more samples are used in the image reconstruction, the structural similarity (SSIM)[2] of the image approaches 1.

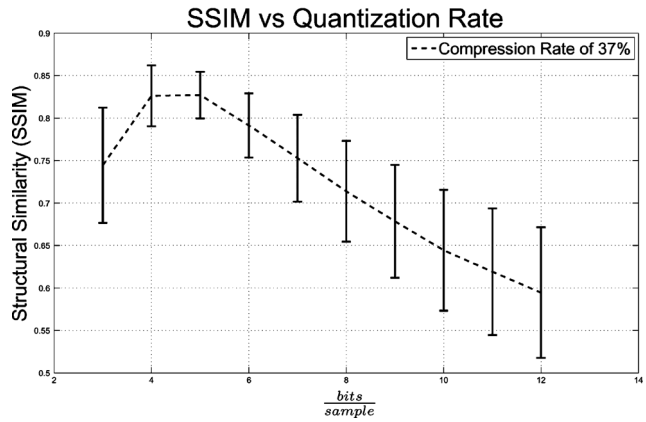


Fig. 5. SSIM vs. quantization bits.

This image distortion can be modeled [41] as

$$\alpha(\gamma) = D_0 - \frac{\Theta}{\gamma - R_0}, \tag{7}$$

where $D_0$, $\Theta$ and $R_0$ are image- or video-dependent constants determined through linear least squares estimation techniques. $\gamma = \frac{M}{N}$ is the user-controlled sampling rate of the image. This allows the sensor node to choose $M$ to balance the benefit of transmitting fewer samples (when $M$ is smaller) with the cost of a decrease in received quality. Note that the function (7) is concave, i.e., the gain in quality achieved by adding more samples diminishes as the total number of samples increases.

*2) Effects of Quantization:* In general, CS theory assumes that the signal is compressed and recovered in the real domain. However, we are usually interested in transmitting a quantized version of the signal. Since the user chooses the value of $M$, which is arbitrary within a certain range, there is a tradeoff between transmitting *fewer samples encoded with more bits each* or transmitting *more samples encoded with fewer bits*. This is examined empirically (again over the images in the SIPI database), and is presented in Fig. 5. It is interesting to note that the highest quality reconstruction occurs when the number of samples per symbol is *lower* than the number of samples per pixel in the original image. This means that there is less precision in

[2]Structural similarity (SSIM) [38] is used to evaluate the quality. The SSIM index is preferred to the more widespread peak signal to noise ratio (PSNR), which has been recently shown to be inconsistent with human eye perception [38]–[40]. SSIM considers three different aspects to determine the similarity between two images. If one image is considered the original, then the measure can be viewed as the relative quality of the second image. The SSIM index first calculates the luminance difference between the two images. Then it subtracts the luminance components out and measures the contrast difference between the two images. Finally, the contrast is divided out and the structural difference is measured as the correlation between the two remaining signals. These three measurements are then combined to result in the overall SSIM index, which is a normalized value between 0 and 1. SSIM is a more accurate measurement of error because the human visual system perceives structural errors in the image more than others. For example, changes in contrast or luminance, although mathematically significant, are very difficult to discern for the human eye. Structural differences such as blurring, however, are very noticeable. SSIM is able to weight these structural differences better to create a measurement closer to what is visually noticeable than traditional measures of image similarity such as mean squared error (MSE) or PSNR. These results have been shown for images [38] and for videos [39], [40] in the LIVE database.

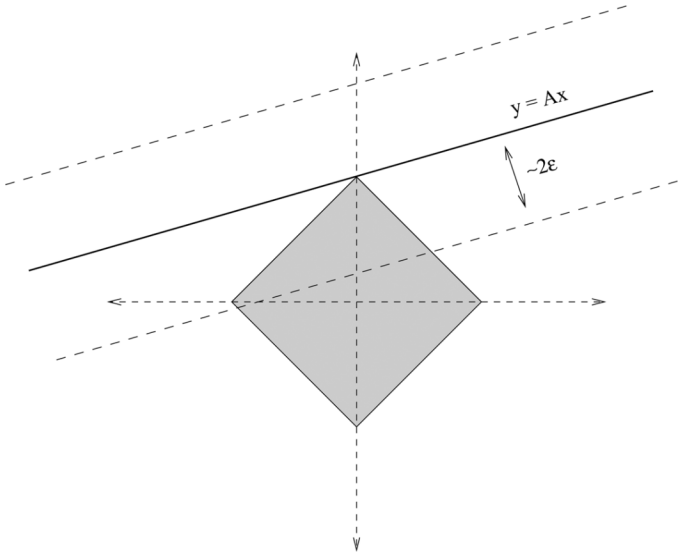


Fig. 6. Geometric interpretation of $\ell_1$ norm minimization.

the samples than in the original pixels, yet we are still able to reconstruct the image with high quality.

This result is in agreement with [42] and also with [13], which shows that CS reconstruction is generally very resistant to low power noise, such as quantization noise. Suppose we have a set of measurement samples $\mathbf{y}^{\#} = \Phi\mathbf{x} + \mathbf{n}$ corrupted by noise, where $\mathbf{n}$ is a deterministic noise term, and is bounded by $\|\mathbf{n}\|_2 < \epsilon$. As long as $\Phi$ obeys (6), then the value of $\mathbf{x}^{\#}$ reconstructed using (4) from $\mathbf{y}^{\#}$ will be within

$$\|\mathbf{x}^{\#} - \mathbf{x}\| \leq C \cdot \epsilon, \tag{8}$$

where $C$ is a "well behaved" constant.[3]

While the full proof of this is beyond the scope of this paper, it is easy to see why $\Phi\mathbf{x}^{\#}$ will be within $2\epsilon$ of $\Phi\mathbf{x}$ using the triangle inequality. Specifically,

$$\|\Phi\mathbf{x}^{\#} - \Phi\mathbf{x}\|_2 \leq \|\Phi\mathbf{x}^{\#} - \mathbf{y}\|_2 + \|\Phi\mathbf{x} - \mathbf{y}\|_2 \leq 2\epsilon. \tag{9}$$

This can be seen graphically in Fig. 6, which represents a system that samples a variable $\mathbf{x} \in \mathbb{R}^2$ with a sampling matrix $A \in \mathbb{R}^{1\times 2}$. The line represents $\mathbf{y} = \Phi\mathbf{x}$, while the diamond represents the $\ell_1$ norm ball. The two dashed lines represent the maximum variation in the samples when corrupted by additive noise of magnitude $\epsilon$. The point where the smallest norm ball intersects the line is the sparsest solution, and is therefore the solution to (4). While this is a simplistic example, it is easy to see that in most cases, the error in the reconstructed sample will result in a small variation in the magnitude of the reconstructed signal. In the scenario represented in Fig. 6, the magnitude of $\epsilon$ would have to be about $\frac{1}{3}$ of the signal power before an incorrect "corner" of the norm ball is selected.

*3) Noise Resilience:* Above we show that the error in the received signal due to quantization is limited to less than a scaled multiple of the error magnitude. While the empirical results shown in Fig. 5 are specific to quantization error, it is clear to see that the analysis presented in (8) and in (9) is *independent of the type of noise*. The only limitation is that the magnitude of the noise is less than $\epsilon$. The results in (8) will hold for any noise distribution where the noise power is less than $\epsilon$ [13], [36]. Any signals reconstructed using (4) will be naturally resilient to errors, *including bit errors from transmission through a noisy channel*.

This is demonstrated empirically in Section V. We show that for a binary symmetric channel with a specific bit error rate (BER), the impact of those errors is negligible as long as the errors are low. While this is also true for traditional video encoders, we can show that the point where the BER becomes high enough that there is visible distortion in the reconstructed image is more than an order of magnitude higher for CVS compared to traditional encoders.

*4) Sampling Complexity:* Traditional image compression schemes partition an image into smaller sections, and compress each of these sections individually. The most well known example of this is in JPEG compression. A JPEG encoder first divides an image into $8 \times 8$ pixel blocks. Then each of these 64 pixel groups are transformed using a DCT transform. JPEG2000 [43] is based on a 2D wavelet transform. However, the actual implementation of that 2D wavelet transform is based on a series of 1D wavelet transforms [44] of each column and row sequentially. Like JPEG, only a portion of the image is processed at a time.

Methods of dividing imaging problems into subproblems are necessary because of the computational complexity required to encode realistic sized images with non-linear transform operations. Like JPEG and JPEG2000, CS imaging must manage this complexity as part of the development of any implementable system. For example, a direct implementation of (2) requires the creation of the $M \times N$ matrix $\Phi$. Assume we are dealing with a $512 \times 512$ pixel image, and that $M$ is set at $\frac{N}{5}$. This will result in a $\Phi$ matrix that is $52{,}429 \times 262{,}144$. A direct implementation would require matrix multiplication with a matrix of over 13 billion elements, which is clearly not practical.

This can be avoided by sampling using a scrambled block Hadamard matrix [45], defined as

$$Y = H_{32} \cdot X, \tag{10}$$

where $Y$ represents image samples (measurements), $H_{32}$ is the $32 \times 32$ Hadamard matrix and $X$ the matrix of the image pixels. The matrix $X$ has been randomly reordered and shaped into a $32 \times \frac{N}{32}$ matrix. Then, $M$ samples are randomly chosen from $Y$ and transmitted to the receiver. The receiver then uses the $M$ samples along with the randomization patterns for both randomizing the pixels into $\mathbf{x}$ (1) and choosing the samples out of $Y$ (both of which can be decided before network setup). The result is a sampling system that is much lower complexity, yet is equivalent to the performance of (2). By reducing the complexity (and therefore the energy) required to compress each frame, we can allocate that energy to encoding more samples (thereby reducing the distortion from the encoder) or increase the energy used to transmit the samples (decreasing the errors caused by the noisy channel). In either case, this reduction in

[3]For practical systems, $C$ is a small constant between 5 and 10 [13].

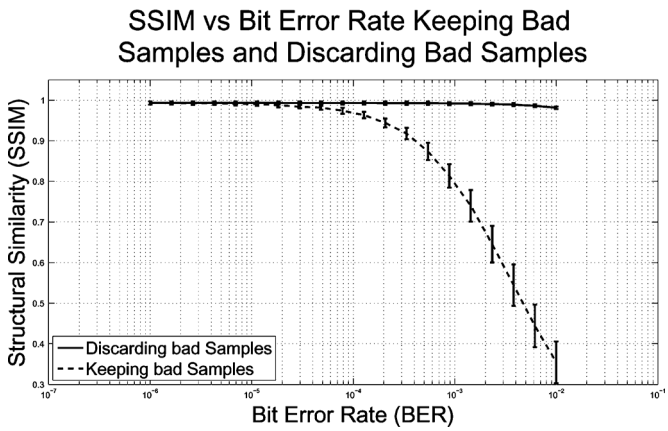


Fig. 7. Compressed Sensed Images Reconstructed With and Without Incorrect Samples.

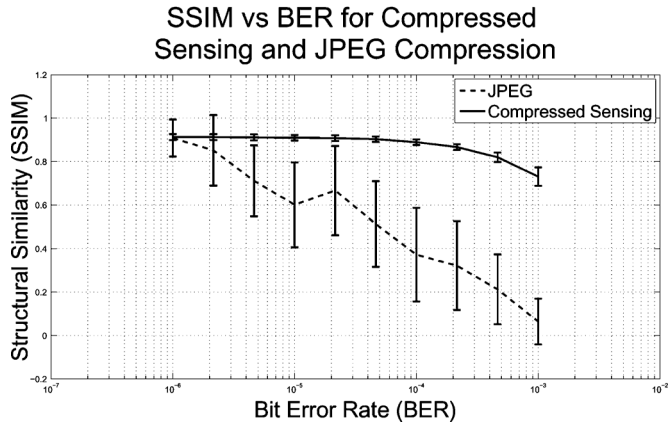


Fig. 8. Structural Similarity (SSIM) vs. Bit Error Rate (BER) for compressed sensed images, and images compressed using JPEG.

encoding complexity can directly lead to an increase in the received quality of the image or video without increasing the required encoding energy.

*5) Wireless Transmission of CS Encoded Images:* CS encoded samples constitute a random, incoherent combination of the original image pixels. This means that, unlike traditional wireless imaging systems, no individual sample is more important for image reconstruction than any other sample. Instead, *the number of correctly received samples* is the main factor in determining the quality of the received image. This naturally leads to a scheme where, rather than trying to correct bit errors, we can instead *detect* errors and simply drop samples that contain errors. This is demonstrated in Fig. 7, where the set of images [37] are encoded using CS and transmitted over a lossy channel. For the purpose of demonstration, we assume that there is a genie at the receiver that is able to perfectly detect when a sample is received incorrectly. We then show the image reconstruction quality with and without those samples. Clearly, simply removing those samples results in a far better reconstruction quality that if those incorrect samples are used in the reconstruction process.

While it is easier to deal with errors in a CS system, the errors that are used in the reconstruction process do not have as much impact on the reconstructed quality as when using a JPEG system. A small amount of random channel errors does not affect the perceptual quality of the received image *at all*, since, for moderate bit error rates, the greater sparsity of the "correct" image will offset the error caused by the incorrect bit. This is demonstrated in Fig. 7. For any BER lower than $10^{-4}$, there is *no noticeable drop in the image quality*. Up to BERs lower than $10^{-3}$, the SSIM is above 0.8, which is an indicator of good image quality. If the BER is kept below $10^{-5}$, there is virtually no distortion in the received image.

This has important consequences and provides a strong motivation for studying compressive wireless video streaming in WMSNs. This inherent resiliency of compressed sensing to random channel bit errors is even more noticeable when compared directly to JPEG. Fig. 8 shows the average SSIM of the SIPI images [37] transmitted through a binary symmetric channel with varying BER. The quality of CS-encoded images degrades gracefully as the BER increases, and is still good for BERs as high as $10^{-3}$. Instead, JPEG-encoded images very quickly deteriorate. This is visually emphasized in Fig. 9, which shows an image taken at the University at Buffalo encoded with CS (above) and JPEG (below) and transmitted with bit error rates of $10^{-5}$, $10^{-4}$, and $10^{-3}$. It is worth pointing out that, since there is some inter-frame prediction in the CS based video encoder as described in Section IV, the errors in the CS encoded images will be propagated to subsequent frames, while the JPEG encoded frame will "refresh" after each frame. However, as can be seen in Fig. 8, on average CS performs much better even with this limitation.

## IV. Compressive Video Sensing (CVS)

While the image encoder described in Section III-B is able to quickly encode a series of images, this only takes advantage of the spatial correlation within each video frame. However, in most video applications there will be significant correlation between consecutive frames. While motion vectors are traditionally used for this, calculating them is a resource intensive operation, and is not appropriate for WMSN implementations. Because of this, we implement a simple algebraic calculation that will compress the data with far lower complexity than traditional methods.

### A. Difference Vector ($\mathbf{dv}$)

Each CVS video frame is designated as either an intra-encoded $I$ frame or a inter- or progressive-encoded $P$ frame. The pattern of the encoded frames is $IPP\cdots PIPP\cdots$, where the distance between two $I$ frames is referred to as the group of pictures (GOP). An $I$ frame is entirely self contained (i.e., the decoder does not need data from any other frame to reconstruct an $I$ frame) and is encoded using (2). A $P$ frame, however, is derived from a previously reconstructed video frame. We do this by calculating the difference vector $\mathbf{dv_i}$ that represents the difference between the $P$ frame samples $\mathbf{y_i}$ and the reference frame samples $\mathbf{y_{ref}}$. For many types of video (i.e., surveillance video), there will be very little difference between consecutive frames and $\mathbf{dv_i}$ can be represented using far fewer bits than $\mathbf{y_i}$.


Fig. 9. Image compressed with CS (above) and JPEG (below) for BER (a) $10^{-5}$, (b) $10^{-4}$, (c) $10^{-3}$.


Fig. 10. Two frames and the difference frame from a video filmed on the State University of New York (SUNY) University at Buffalo's North Campus. (a) $x_{ref}$, (b) $x_i$, (c) $|\Delta x_i|$.

Formally, the difference vector is the difference between two encoded frames,[4] and is denoted as

$$\begin{aligned}\mathbf{dv_i} &= \mathbf{y_i} - \mathbf{y_{ref}} \\ &= \Phi\mathbf{x_i} - \Phi\mathbf{x_{ref}} \\ &= \Phi(\mathbf{x_i} - \mathbf{x_{ref}}).\end{aligned} \tag{11}$$

This shows that, while the source can only sample the frames after encoding, this is equivalent to sampling the difference between the two frames explicitly (assuming $\Phi$ stays constant). We can then reconstruct $\mathbf{x_i}$ at the receiver by using the received version of $\hat{\mathbf{dv}}_\mathbf{i}$ in the optimization problem

$$\begin{aligned}\underset{\mathbf{s}}{\text{minimize}} \quad & \|\mathbf{s}\|_1 \\ \text{subject to} \quad & \|\mathbf{y_{ref}} + \hat{\mathbf{dv}}_\mathbf{i} - \tilde{\Psi}\mathbf{s}\|_2^2 < \epsilon.\end{aligned} \tag{12}$$

As long as the error between $\mathbf{y_{ref}} + \hat{\mathbf{dv}}_\mathbf{i}$ and $\mathbf{y_i}$ is "small enough", the reconstructed image will be very close to the original image. In many ways, this is similar to the work presented in [47]. However, unlike [47], CVS uses the *entire* frame, which will increase the sparsity and allow us to use CS sampling on the difference vector explicitly.

The advantage of transmitting $\mathbf{dv_i}$ instead of $\mathbf{y_i}$ is that, due to temporal correlation, $\mathbf{dv_i}$ can be compressed much more than $\mathbf{y_i}$. To see this, we first examine the properties of the difference frame $\Delta\mathbf{x_i} = \mathbf{x_i} - \mathbf{x_{ref}}$. For stationary camera applications, the difference between two consecutive (or nearly consecutive) frames will be small. This is presented graphically in Fig. 10, which shows two consecutive frames from a video filmed on the State University of New York (SUNY) University at Buffalo's North Campus, along with the frame representing the difference[5] between the two frames.

Quantitatively, this corresponds to a reduction in the standard deviation of the symbols, resulting in a decrease in the number of bits required to represent the images. The standard deviation of the frames in Fig. 10 and the standard deviation of the DCT transform of those frames is shown in Table I. This allows us to use fewer quantization bits to represent each frame without causing any additional distortion to the signal. This is presented graphically in Fig. 11 for the frames shown in Fig. 10.

While it is clear from Fig. 11 that the difference frame has less variation than the original frame, it is more important to

[4]The encoded frames are used in this case to avoid the necessity of capturing the entire video at the sensor node. This allows the sensors to use something such as a single pixel camera [46]

[5]For clarity, since the $\mathbf{dv}$ can take both positive and negative values, the magnitude of each pixel is displayed.

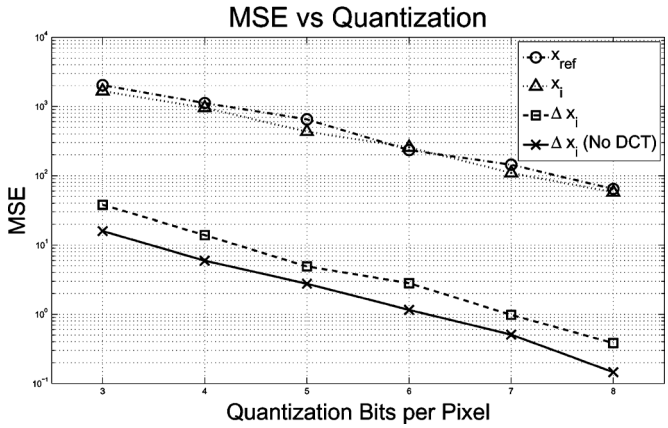


Fig. 11. MSE Distortion from Quantization for Varying Bits-per-Pixel.

TABLE I
STANDARD DEVIATION OF VIDEO FRAMES

| Frame | $x_{ref}$ | $x_i$ | $\Delta x_i$ |
|---|---|---|---|
| $\sigma^2$ of Frame | 67.84 | 67.30 | 18.16 |
| $\sigma^2$ of DCT of Frame | 140.56 | 140.84 | 18.17 |

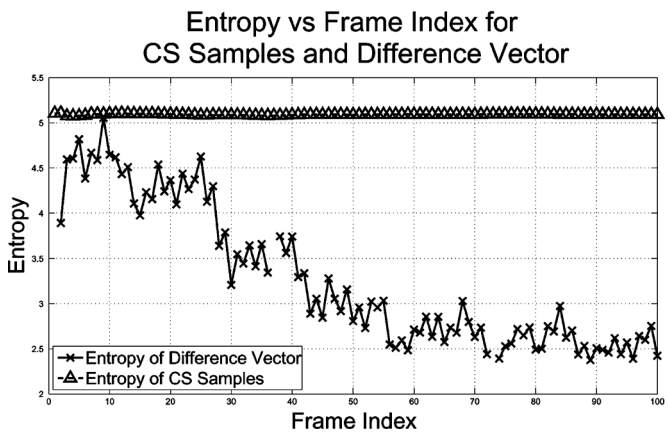


Fig. 12. Entropy per Frame Index for Samples $\mathbf{y_i}$ and Difference Vector $\mathbf{dv_i}$.

see how this affects the compression of $\mathbf{dv_i}$ (as defined in (11)). We first note that $\mathbf{y_i}$ is a compressed version of $\mathbf{x_i}$. Looking at the difference vector between $\mathbf{y_i}$ and $\mathbf{y_{ref}}$, we would assume that the entropy of $\mathbf{dv_i}$ should be lower than the entropy of $\mathbf{y_i}$. We can see in Fig. 12 that this is indeed the case for a typical surveillance video, where the entropy of $\mathbf{y_i}$ and $\mathbf{dv_i}$ for the first 100 frames of the video are shown. After some initial large scale motion for the first few seconds of video, the entropy of the dv is significantly lower than that of the original samples. What is important to note is that $\mathbf{dv_i}$ is based on an *already compressed frame*, and that any further compression is in addition to the compression of the original samples.

It is shown in [13] that if CS (as in Section III-A) is used to compress $\mathbf{dv_i}$, a decrease in $M$ will cause the smaller components of $\hat{\mathbf{dv_i}}$ to get forced to zero (where $\hat{\mathbf{dv_i}}$ is the version of $\mathbf{dv_i}$ reconstructed using (4)). The reconstructed $\hat{\mathbf{dv_i}}$ will be very close to $\mathbf{dv_i}$ for the components of $\mathbf{dv_i}$ where the magnitude is large (i.e., when there is a large difference between $\hat{\mathbf{y}}_\mathbf{i}$ and $\mathbf{y_{ref}}$) and zero for the small (or already zero) components.

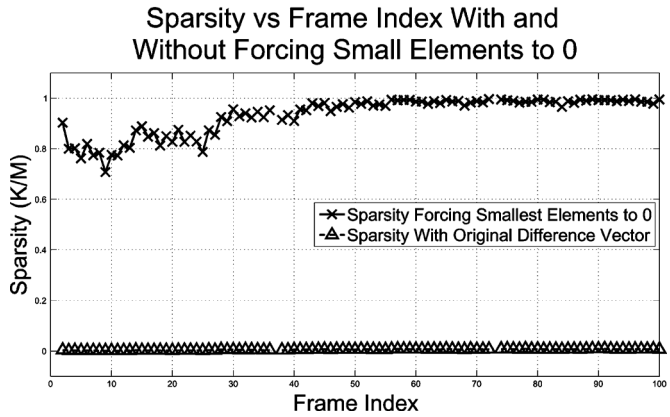


Fig. 13. Sparsity vs. Frame Index With and Without Forcing Small Elements to zero.

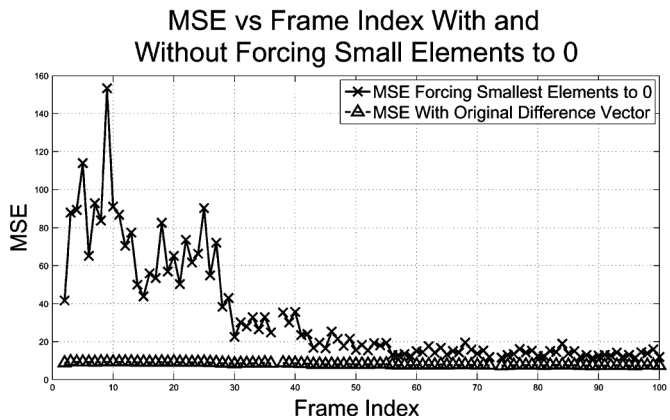


Fig. 14. MSE vs. Frame Index With and Without Forcing Small Elements to zero.

This is acceptable as long as the effect of the small components getting forced to zero on the reconstructed signal $\hat{\mathbf{dv_i}}$, and then on the CS vector $\hat{\mathbf{y}}_\mathbf{i} = \mathbf{y_{ref}} + \hat{\mathbf{dv_i}}$, is acceptable.

We show in Fig. 13 and in Fig. 14, that, for a video filmed at the University at Buffalo, this is indeed the case. Fig. 13 shows the sparsity of the reconstructed signal when the small components (those that are within a single quantization level) are set to zero. The mean squared error between $\hat{\mathbf{y}}_\mathbf{i}$ and $\hat{\mathbf{dv_i}}$ for each frame is shown in Fig. 14. First we see in Fig. 13 that, while few of the components of $\mathbf{dv_i}$ are exactly zero, most of them are "close" to zero, showing that, for reasonable sampling rates, the large components of $\mathbf{dv_i}$ will be retained in $\hat{\mathbf{dv_i}}$.

We further see in Fig. 14 that, while the MSE in the reconstructed sampling vector $\hat{\mathbf{y}}_\mathbf{i}$ is higher then when the entire $\mathbf{dv_i}$ is used (including the small components), the MSE is at all times below the level that would cause visual distortion in the reconstructed video frame. Setting the samples to zero does increase the MSE, notably in the first few seconds that contain high motion, but at all times it is below the MSE level that would cause noticeable error in the reconstructed image. These results agree with the observation of the entropy of these frames, and demonstrate that, when there is very high correlation between frames, this system can be used to significantly decrease the number of samples needed to represent an image.

## V. Energy-Rate-Distortion of Video Encoders in Unreliable Channels

Comparing the performance of video encoders is not a straightforward task. It is important to consider metrics that are relevant to the scenario and environment in which the proposed system will be deployed. If we simply consider (as is typical) rate distortion performance, then CVS performs on par with MJPEG, and significantly below more complex encoders such as H.264/AVC. However, to evaluate the performance of CVS *in a WMSN* (or any energy-constrained environment), we need to be able to accurately model the performance of the encoder, along with traditional reference encoders, in an energy constrained environment taking encoding energy consumption into account. In this section, we develop a model (in terms of SSIM [38]) for video that has been encoded and transmitted over a WMSN. We are interested in modeling the received video quality as a function of the encoded video rate $r_v$, the channel coding rate $r_{ch}$, the total energy budget $E_B$ per frame and the channel quality.

### A. Video Encoders Used for Comparison

Before presenting the energy-rate-distortion analysis, we will first briefly introduce the two traditional video encoders used for comparison.

*1) Motion-JPEG (MJPEG) Video Encoder:* MJPEG video encoding is an intra-frame encoding scheme based on the JPEG image compression standard [48]. Though there is no official standardization of MJPEG, the basic concepts of most implementations are the same. Each frame is first divided into $8 \times 8$ blocks which are transformed to the frequency domain using the discrete cosine transform (DCT) [26], creating an $8 \times 8$ block of DCT coefficients. The DCT coefficients of each macroblock are then quantized and entropy encoded, resulting in a much smaller file than the original. The resulting video has compression and quality comparable to JPEG image compression and can be done without significant complexity requirements at the encoder. These factors have made this protocol very useful in low complexity devices such as digital cameras.

*2) H.264/AVC Intra Video Encoder:* H.264/AVC intra video compression represents the state of the art in current video compression techniques. The basic functionality of the H.264/AVC intra [8] encoder is similar to that of JPEG with the major addition of intra-prediction. Along with the frequency transform—quantization—entropy encoding functionalities, the encoder will take an image block and compare it to other macroblocks either within the same frame (intra-prediction) or in a previous frame (inter-prediction). The previously decoded blocks are used to predict the current block. As the information needed to indicate the prediction is much less than what is needed to encode the block explicitly, a significant amount of compression can be gained using such a method. However, the tradeoff of this is the complexity needed to find the reference macroblocks.

By finding the difference between two macroblocks and encoding that difference (which is generally very small), the encoder can greatly reduce the amount of data necessary to represent a video at a very good quality. For a full explanation of H.264/AVC intra video encoding, the reader is referred to [7]–[9].

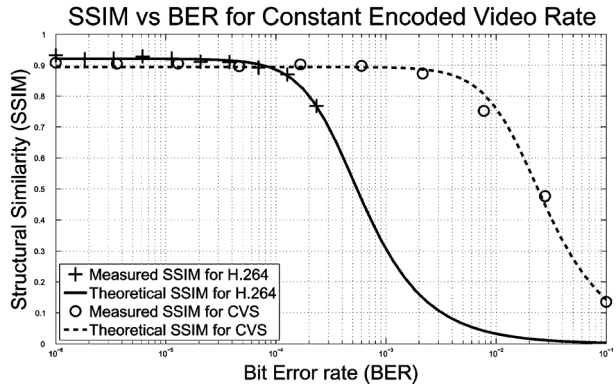


Fig. 15. SSIM vs. BER for H.264/AVC intra and CVS Encoders.

### B. Energy-Rate-Distortion Model

To analyze the rate distortion performance of video encoders, we must first develop a model that accurately predicts the effect of compression and bit errors on the video quality. In a lossless channel, video distortion can be modeled [41] as

$$\alpha(r_v) = D_0 - \frac{\Theta}{r_v - R_0}, \tag{13}$$

where $D_0$, $\Theta$ and $R_0$ are video dependent constants determined through linear least squares estimation techniques.

Though this model works very well when there are no errors, any bit errors can decrease the quality of the received video. Unlike typical data networks, however, the video does *not* have to be received perfectly for it to be acceptable at the user. This can be seen by observing a plot of the received video quality as a function of the bit error rate of the received video, as is shown in Fig. 15. For this plot, the videos were encoded to an acceptable quality,[6] transmitted through a binary symmetric channel with varying bit error rates (BER) and then decoded. For low BER, there is almost no affect on the received SSIM. As the BER increases, however, the video quality drops off significantly.

Based on this observation, we have modeled the error performance as a low pass filter using

$$U(r_{ch}, r_v) = \frac{\alpha(r_v)}{\sqrt{1 + \tau^2 (BER(r_{ch}, r_v))^2}} \tag{14}$$

where $r_{ch}$ is the channel coding rate (in $\frac{bits\ in}{bits\ out}$), $r_v$ is the encoded video rate in kbit/s and $U(r_{ch}, r_v)$ is the quality of the received video in SSIM as a function of $r_{ch}$ and $r_v$. The encoder dependent constant $\tau$ is used to indicate the BER level of the quality dropoff.

This model was chosen over other available models (i.e., [41]) because, not only did it fit the experimental data better than other available models, but it emphasizes the two factors that affect the received video quality in this system. Specifically, the encoding rate and the channel error rate. In many ways, the equation can be viewed as a low pass filter. The numerator, $\alpha(r_v)$, essentially limits the maximum "gain" of the

[6]For this work, "acceptable" quality is defined as $\mathrm{SSIM} \geq 0.8$.

system. This makes sense intuitively because the quality of the video at output of the video encoder (i.e., before wireless transmission) is the *maximum* quality obtainable from the system. Experimental results also show that, while video streams are generally very resilient to bit errors, there comes a *maximum BER* after which the quality decreases dramatically, which supports the "low-pass filter like" properties, and allows us to focus on the BER where the quality begins to drop off, which, as we will show below, is the optimal operating point from an energy consumption perspective.

### C. SNR Model

Consider the energy budget per frame $E_B$ as the energy available to the system during each frame period $t_f = \frac{1}{fps}$, where $fps$ represents the number of frames per second of the video. We can then express the average energy required for video encoding as

$$E_E(r_v) = E_{E,max} \cdot t_e(r_v), \tag{15}$$

where $E_{E,max}$ is the maximum energy available to the encoder during the frame period, and $t_e(r_v)$ is the processor load, i.e., the time fraction of a frame that the encoder needs to encode video at rate $r_v$. We will explain how this is used explicitly in more detail below, but calculating $E_E$ as in (15) will allow us to determine the amount of energy required for encoding *without including energy for background processing*.

We then look at the energy required to transmit a video frame.[7] The transmitted energy per video frame $E_T$ is defined as

$$E_T = (E_B - E_E(r_v)), \tag{16}$$

i.e., the total energy available reduced by the energy needed to encode the video.

For the encoders considered in this paper, the empirical models

$$t_e(r_v) = a\, r_v + b, \tag{17}$$

and

$$t_e(r_v) = c - \frac{T}{r_v + d}, \tag{18}$$

accurately model the processor load as a function of the encoded video rate, as shown in Fig. 16 where $a$, $b$, $c$, $d$ and $T$ are platform dependent positive constants determined through linear regression analysis of the encoder implementation.

To obtain these results, all three encoders are run at all available encoded video rates on the same platform. For this model, the platform is an Intel Core 2 Duo processor running Ubuntu 10.10. The time $t_f = \frac{1}{fps}$, defined as the inverse of the framerate of the video, is used as the maximum allowed encoding time, i.e., the mean encoding time per frame for a real time video *must* be less than $t_f$. The actual encoding time per frame, $t_v$ is measured or estimated and compared to $t_f$. We can then find the value $t_e = \frac{t_v}{t_f}$ which represents the fraction of time used to encode each frame. This allows us to measure the time

[7]For now, we look at energy-per-frame as opposed to energy-per-bit so that the comparison between encoding energy and transmission energy is clear. We will convert the energy-per-frame calculations to energy-per-bit when the actual SNR is calculated later in this section.

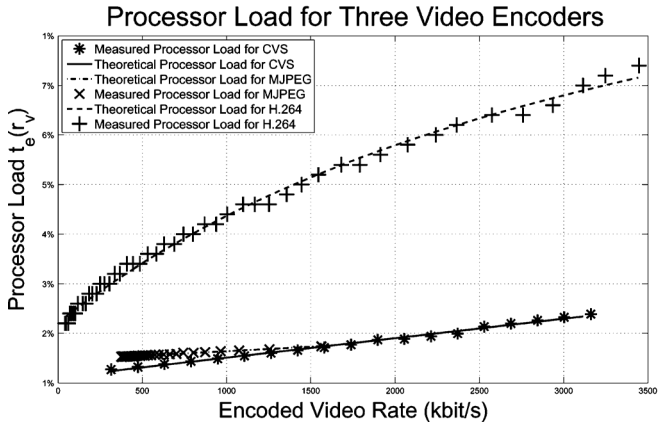


Fig. 16. Processor Load vs. Encoded Video rate.

required for encoding video. If we look at two energy states, $E_{high}$ (or $E_{E,max}$, as defined in (15)) for the energy consumption when processing video, and $E_{bkgnd}$ for the energy used for background processing, this will allow us to determine how much energy is spent on encoding video compared to how much energy the processor is using to turn on perform basic functions.

For example, if a 30 fps video of 3000 frames takes 15 seconds to encode (i.e., 200 fps encoding time) at some rate $r_{v_0}$, we find that each frame takes $t_v = \frac{1}{200}$ of a second to encode. In the video, each frame lasts $t_f = \frac{1}{30}$ of a second. This means that on average, for each frame, $t_e = \frac{30}{200} = 15\%$ of the frame time is needed to encode each frame. Taking the maximum encoder energy use per frame as 0.5 J (the value for the system used to test), then it will take on average $0.5 \times 15\% = 75$ mJ to encode that video at $r_{v_0}$.

We can then give the SNR model as

$$SNR(r_{ch}, r_v) = \frac{L \cdot r_{ch} \cdot d_{free} \cdot (E_B - E_E(r_v))}{N_0 \frac{r_v}{r_{ch} \cdot fps}}, \tag{19}$$

where $L$ is the path loss, $N_0$ is the noise power and $d_{free}$ is the free distance of the channel code $r_{ch}$. The denominator term $\frac{r_v}{r_{ch} \cdot fps}$ is used to change from energy-per-frame to energy-per-bit. As $r_v$ increases, the energy needed to encode the video increases while the transmission energy per bit decreases, causing the SNR to decrease.

## VI. Energy-Rate-Distortion Optimization

In Section V, we developed a set of equations to model the received video quality after transmission through a noisy channel with a finite energy budget per frame. In this section, we use this model to find the video encoder that results in the highest received video quality as a function of the energy budget $E_B$. We first seek to find the optimal allocation of $r_{ch}$ and $r_v$ for a fixed energy budget. By repeating this for multiple values of $E_E$, we can then develop a model of the *optimal* received video quality as a function of $E_B$. The goal of this optimization is to determine i) which encoder requires the lowest energy to achieve a target video quality, and ii) what values of $r_{ch}$ and $r_v$ should be chosen to obtain that quality. While we are optimizing $r_{ch}$ and $r_v$ explicitly, this will in turn select the allocation of $E_B$ into $E_E$ and $E_T$. This is because, as shown in the previous section,

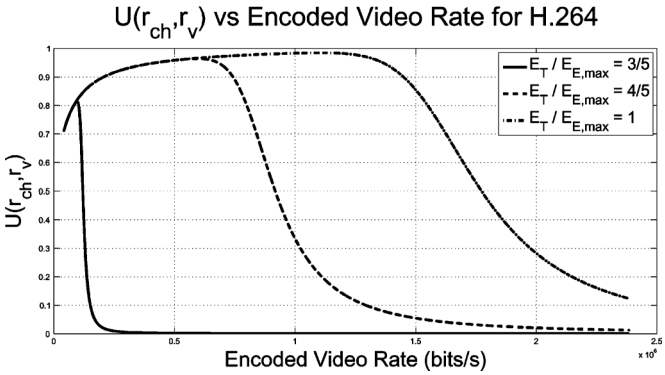


Fig. 17. Utility Function vs. Encoded Video Rate for H.264/AVC intra.

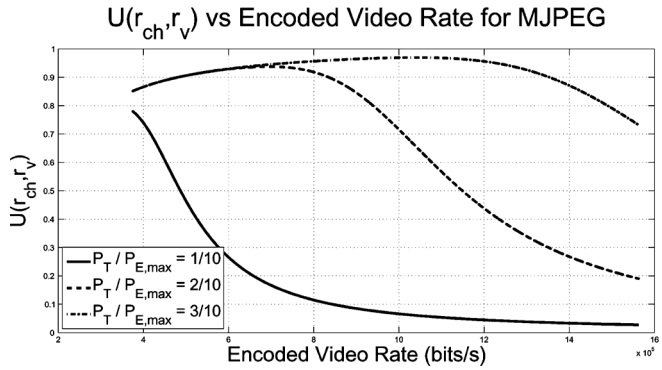


Fig. 19. Utility Function vs. Encoded Video Rate for MJPEG.

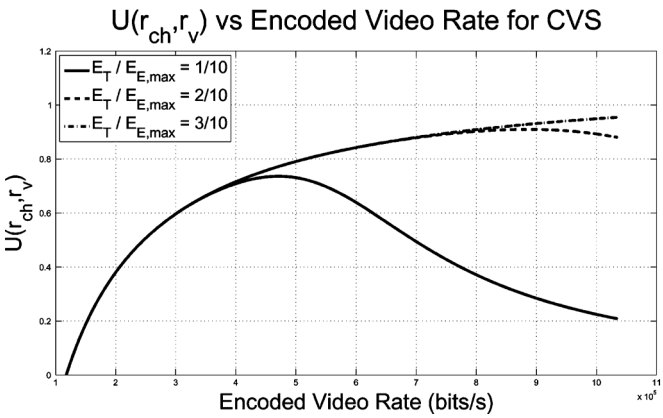


Fig. 18. Utility Function vs. Encoded Video Rate for CVS.

the amount of energy required to encode the video is dependent on the encoding rate, as is seen in (17) and (18). In addition, the SNR is dependent on the number of bits transmitted, which is clearly directly dependent on both $r_{ch}$ and $r_v$.

We begin by holding $E_B$ constant and finding the optimal allocation of $r_{ch}$ and $r_v$. This problem can be modeled as the optimization problem

$$\begin{aligned} \underset{r_{ch},r_v}{\text{maximize}} \quad & U(r_{ch}, r_v) \\ \text{subject to} \quad & E_B \geq E_E(r_v) + E_T, \end{aligned} \tag{20}$$

where $E_T$ is the energy available for transmission. Based on the analysis in Section V, we can formulate the problem as

$$\begin{aligned} \underset{r_{ch},r_v}{\text{maximize}} \quad & \frac{D_0 - \frac{\Theta}{r_v - R_0}}{\sqrt{1 + \left(\tau \cdot Q\left(\sqrt{SNR(r_{ch}, r_v)}\right)\right)^2}} \\ \text{subject to} \quad & E_B \geq E_E(r_v) + E_T. \end{aligned} \tag{21}$$

The solution to this problem gives us the optimal channel rate and video encoded rate for a given energy budget and a given encoder. Plots of the objective function for H.264/AVC intra, CVS and MJPEG are presented in Figs. 17, 18 and 19 showing the optimal rates for the given energy values. These are plotted for different values of $\frac{E_T}{E_{E,max}}$, which is the ratio of the total energy budget to the maximum energy per frame. For clarity, the function as plotted is actually $U(r^*_{ch}, r_v)$ vs. $r_v$, where $r^*_{ch}$ is the value of $r_{ch}$ that maximizes $U(r_{ch}, r_v)$ for the given value of $r_v$.

The maximum value of this objective function is the optimal video quality, and the values of $r_v$ and $r_{ch}$ that achieve that point are the optimal encoding rates. However, because the Q error function has no closed form solution, the problem must be solved numerically (though the Q error function can be approximated for some values of SNR, the resulting optimization problem is still a non-linear, non-convex discrete program without any obvious solution). To simplify the analysis, note that in all cases, the quality of the received video follows the same pattern. The"filter-like" form of (14) results in a very sharp decrease in the quality after the rate increases beyond a certain point. It is safe to assume that this dramatic drop-off is due to the BER at that rate increasing beyond the cutoff point, driving the received video quality down.

Based on this, we assume that the optimal value would be obtained by a video encoder and channel encoder rate that are very close to that cutoff point. This leads to the much simpler optimization problem

$$\underset{r_{ch}}{\text{minimize}} \quad \left| Q\left(\sqrt{SNR(r_{ch}, r_v)}\right) - \frac{1}{2\pi\tau} \right|^2 \tag{22}$$

which states that the optimal point is the one that causes the BER to be as close as possible to that cutoff. This analysis reduces the original two dimensional optimization problem (20) to an optimization problem over a single dimension. For practical channel coders [49], the length of $r_{ch}$ is generally less than 10. In comparison, the length of $r_v$ can be 30 (MJPEG), 50 (H.264/AVC intra) or $r_v$ can be continuous (CVS). By removing the search over $r_v$, we are reducing the majority of the search space of the problem, which will greatly reduce the complexity.

Simple tests show that in all cases, the values obtained from (22) are close to the optimal solution. The maximum error in SSIM was 0.31% for CVS, 1.12% for MJPEG and 0.94% for H.264/AVC intra.

## VII. Performance Evaluation

The objective of the optimization problem (20) or the simplified optimization problem (22) is two-fold. First, it allows comparing the performance of different video encoders. Second, once the optimal encoder is found, it finds the optimal values for

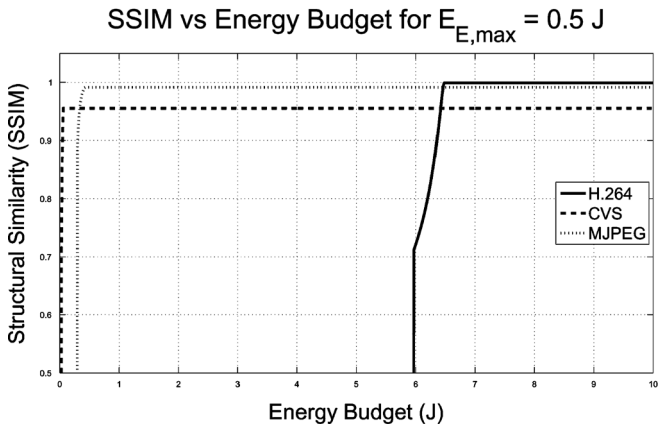


Fig. 20. SSIM vs. Total Energy Budget for $E_{E,max} = 0.5J$.

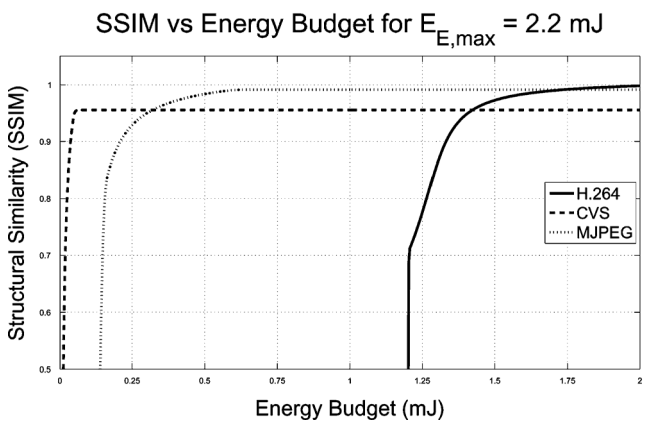


Fig. 21. SSIM vs. Total Energy Budget for $E_{E,max} = 2.2mJ$.

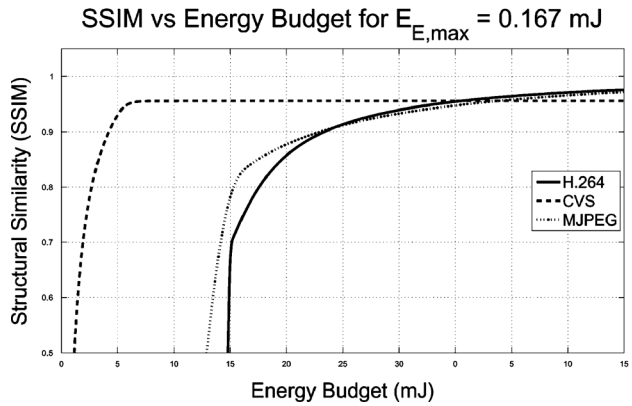


Fig. 22. SSIM vs. Total Energy Budget for $E_{E,max} = 0.167mJ$.

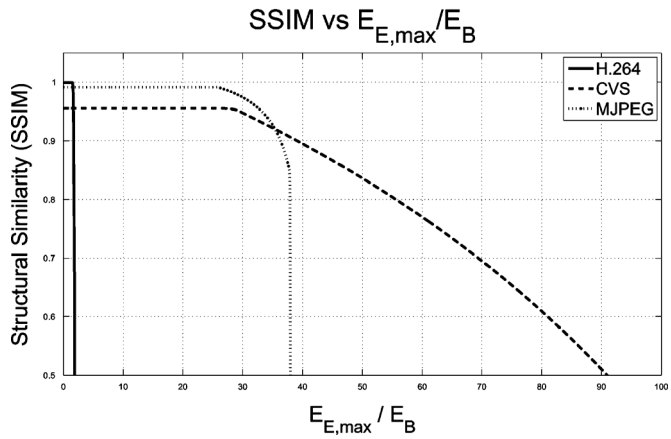


Fig. 23. SSIM vs. $\frac{E_{E,max}}{E_T}$.

the encoded video rate and the channel encoder rate that result in the optimal performance.

### *A. Analyzing Different Encoders*

A major advantage of the analysis presented in the previous sections is that it is independent of any specific platform or encoder. To compare the performance of different encoders, we explore the design space varying the values for noise power, path loss, and $E_{E,max}$ of the system. To determine the optimal encoder for a specific platform, we empirically determine the energy-rate performance for the platform, and the rate distortion performance for the type of video being encoded.

Below we give example plots with different processors resulting in different values for $E_{E,max}$. First, we consider the case where the video originates at a relatively high powered system with a maximum encoding energy cost of 0.5 J (i.e., the energy to encode a frame on a desktop or laptop computer), and is shown in Fig. 20. Even though the higher power system is able to encode video faster, the limiting factor in this system is the energy required to encode at *any* quality. Once encoding is possible, the SNR required to achieve a "good" quality received video is easily achieved. The second two situations are shown in Figs. 21 and 22. These two plots are generated with maximum encoding cost of 2.2 mJ (the energy to encode a frame on a smart phone) and 0.167 mJ (the energy to encode a frame on a small sensor node) respectively. These values were chosen to represent smaller platforms that have significantly lower processor energy requirements.

In all of these simulations, there is a tradeoff between energy and received video quality. The CVS encoder results in a lower maximum received video quality, but can generally achieve that max quality at a much lower energy requirement than either MJPEG or H.264/AVC intra. For example say we want to achieve a 0.8 SSIM ("good" quality) with a maximum encoding cost of 2.2 mJ, as shown in Fig. 21. We can see that the CVS encoder crosses the 0.8 SSIM level very close to 0 mJ. The MJPEG encoder crosses around 0.15 mJ while the H.264/AVC intra encoder crosses at 1.25 mJ. This means that *we can achieve the same quality for much lower energy cost using CVS*. Clearly, the analysis is dependent on the noise power, path loss, encoder implementation and other application specific factors.

To get a more general comparison, Fig. 23 shows the achievable received video quality as the relative ratio of maximum encoder energy to total energy budget is increased. This allows us to view the optimal received video quality without the dependency on a specific platform. Because of its low encoding cost, CVS is able to achieve good video quality even when the cost of encoding the video increases. Since H.264/AVC intra needs more energy to encode the video, it is unable to produce a video when the relative cost of encoding becomes too high.

## VIII. Conclusions and Future Work

We have presented a compressed sensing based video encoder designed to encode and transmit video in wireless multimedia sensor networks. We use a novel rate-energy-distortion analysis to compare the video transmission over wireless links with a limited energy budget for low-complexity sensing devices. CVS is compared to two common video encoders; H.264/AVC intra and MJPEG. It can be seen that CVS outperforms H.264/AVC intra and MJPEG when the encoding energy is high compared to the video transmission energy. However, when energy is not as restricted, H.264/AVC intra can achieve better video quality because of its better rate distortion performance.

## References


[1] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Comput. Netw. (Elsevier)*, vol. 51, no. 4, pp. 921–960, Mar. 2007.

[2] S. Pudlewski and T. Melodia, "A rate-energy-distortion analysis for compressed-sensing-enabled wireless video streaming on multimedia sensors," in *Proc. IEEE Global Communications Conf. (GLOBECOM)*, Houston, TX, USA, Dec. 2011.

[3] S. Soro and W. Heinzelman, "A survey of visual sensor networks," *Adv. Multimedia*, vol. 2009, 2009, Article ID 640386.

[4] A. Kansal, S. Nath, J. Liu, and F. Zhao, "SENSE-WEB: An infrastructure for shared sensing," *IEEE MultiMedia*, vol. 14, no. 4, pp. 8–13, Oct. 2007.

[5] A. T. Campbell, N. D. Lane, E. Miluzzo, R. Peterson, H. Lu, X. Zheng, M. Musolesi, K. Fodor, S. B. Eisenman, and G. S. Ahn, "The rise of people-centric sensing," *IEEE Internet Comput.*, vol. 12, no. 4, pp. 12–21, Jul./Aug. 2008.

[6] Y. Gu, Y. Tian, and E. Ekici, "Real-time multimedia processing in video sensor networks," *Signal Process.: Image Commun. J. (Elsevier)*, vol. 22, no. 3, pp. 237–251, Mar. 2007.

[7] Advanced Video Coding for Generic Audiovisual Services, ITU-T Recommendation H.264, 2005.

[8] T. Wiegand, G. J. Sullivan, G. Bjntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits and Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[9] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammar, and T. Wedi, "Video coding with H.264/AVC: Tools, performance, and complexity," *IEEE Circuits Syst. Mag.*, vol. 4, no. 1, pp. 7–28, Apr. 2004.

[10] T. Wiegand, G. J. Sullivan, J. Reichel, H. Schwarz, and M. Wien, "Joint draft 11 of SVC amendment," *Doc. JVT-X201*, Jul. 2007.

[11] D. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[12] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[13] E. J. Candes, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Commun. Pure Appl. Math.*, vol. 59, no. 8, pp. 1207–1223, Aug. 2006.

[14] S. Pudlewski, T. Melodia, and A. Prasanna, "Compressed-sensing-enabled video streaming for wireless multimedia sensor networks," *IEEE Trans. Mobile Comput.*, vol. 11, no. 6, pp. 1060–1072, June 2012.

[15] S. Pudlewski and T. Melodia, "On the performance of compressive video streaming for wireless multimedia sensor networks," in *Proc. IEEE Int. Conf. Communications (ICC)*, Cape Town, South Africa, May 2010.

[16] Z. He, Y. Liang, L. Chen, I. Ahmad, and D. Wu, "Power-rate-distortion analysis for wireless video communication under energy constraints," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 5, pp. 645–658, May 2005.

[17] Z. He, W. Cheng, and X. Chen, "Energy minimization of portable video communication devices based on power-rate-distortion optimization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 5, pp. 596–608, May 2008.

[18] F. Bellard [Online]. Available: http://www.ffmpeg.org

[19] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proc. IEEE*, vol. 93, no. 1, pp. 71–83, Jan. 2005.

[20] L. W. Kang and C. S. Lu, "Distributed compressive video sensing," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, Taipei, Taiwan, Apr. 2009, pp. 1169–1172.

[21] M. Wakin, J. Laska, M. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. Kelly, and R. Baraniuk, "An architecture for compressive imaging," in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, Oct. 2006, pp. 1273–1276.

[22] M. Wakin, J. Laska, M. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. Kelly, and R. Baraniuk, "Compressive imaging for video representation and coding," in *Proc. Picture Coding Symp. (PCS)*, Beijing, China, Apr. 2006.

[23] T. Do, Y. Chen, D. Nguyen, N. Nguyen, L. Gan, and T. Tran, "Distributed compressed video sensing," in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, Nov. 2009, pp. 1393–1396.

[24] V. Stankovic, L. Stankovic, and S. Cheng, "Compressive video sampling," in *Proc. Eur. Signal Processing Conf. (EUSIPCO)*, Lausanne, Switzerland, Aug 2008, pp. 2–6.

[25] A. Graps, "An introduction to wavelets," *IEEE Computat. Sci. Eng.*, vol. 2, no. 2, pp. 50–61, 1995.

[26] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. C-23, no. 1, pp. 90–93, Jan. 1974.

[27] A. Bruckstein, D. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Rev.*, vol. 51, no. 1, pp. 34–81, Feb. 2007.

[28] J. Romberg, "Imaging via compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 14–20, Mar. 2008.

[29] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, Mar. 2004.

[30] I. E. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*. Philadelphia, PA, USA: SIAM, 1994.

[31] M. Zhu and T. Chan, An Efficient Primal-Dual Hybrid Gradient Algorithm for Total Variation Image Restoration, tech. rep., UCLA CAM Report 08-34, 2008.

[32] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *J. Roy. Statist. Soc. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[33] 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 J. Select. Topics Signal Process.*, vol. 1, no. 4, pp. 586–598, 2007.

[34] J. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2231–2242, Oct. 2004.

[35] D. L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, Sparse Solution of Underdetermined Linear Equations by Stagewise Orthogonal Matching Pursuit, Stanford Tech. Rep., 2006.

[36] M. Salman Asif and J. Romberg, "Dynamic updating for $ell_1$ minimization," *IEEE J. Select. Topics Signal Process.*, vol. 4, no. 2, pp. 421–434, Apr. 2010.

[37] USC Signal and Image Processing Institute [Online]. Available: http://sipi.usc.edu/database/index.html

[38] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[39] S. S. Hemami and A. R. Reibman, "No-reference image and video quality estimation: Applications and human-motivated design," *Signal Process.: Image Commun.*, vol. 25, no. 7, pp. 469–481, 2010.

[40] S. Chikkerur, V. Sundaram, M. Reisslein, and L. J. Karam, "Objective video quality assessment methods: A classification, review, and performance comparison," *IEEE Trans. Broadcast.*, vol. 57, no. 2, pp. 165–182, Jun. 2011.

[41] K. Stuhlmuller, N. Farber, M. Link, and B. Girod, "Analysis of video transmission over lossy channels," *IEEE J. Select. Topics Signal Process.*, vol. 18, no. 6, pp. 1012–1032, Jun. 2000.

[42] M. Li, S. Rane, and P. Boufounos, "Quantized embeddings of scale-invariant image features for mobile augmented reality," in *Proc. IEEE Int. Workshop Multimedia Signal Processing (MMSP)*, Sep. 2012, pp. 1–6.

[43] JPEG2000 Requirements and Profiles, ISO/IEC JTC1/SC29/WG1 N1271, Mar. 1999.

[44] W. Sweldens, "The lifting scheme: A new philosophy in biorthogonal wavelet constructions," in *Wavelet Applications in Signal and Image Processing III*, A. F. Laine and M. Unser, Eds., 1995, pp. 68–79, Proc. SPIE 2569.

[45] L. Gan, T. Do, and T. D. Tran, "Fast compressive imaging using scrambled block Hadamard ensemble," in *Proc. Eur. Signal Processing Conf. (EUSICPO)*, Lausanne, Switzerland, 2008.

[46] M. Duarte, M. Davenport, D. Takhar, J. Laska, T. Sun, K. Kelly, and R. Baraniuk, "Single-pixel imaging via compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 83–91, Mar. 2008.
[47] S. Mun and J. Fowler, "Residual reconstruction for block-based compressed sensing of video," in *Proc. IEEE Int. Data Compression Conf. (DCC)*, Mar. 2011, pp. 183–192.
[48] "Digital compression and coding of continuous-tone still images—Requirements and guidelines," ITU-T Recommendation T.81, 1992.
[49] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Trans. Commun.*, vol. 36, no. 4, pp. 389–400, Apr. 1988.

**Scott Pudlewsk** is a Technical Staff Member in the Wireless Tactical Networking Group at the Massachusetts Institute of Technology, Lincoln Laboratory. He received his Ph.D. in electrical engineering from the State University of New York (SUNY) at Buffalo in 2012. He previously received his M.S. from SUNY Buffalo in 2010 and his B.S. from the Rochester Institute of Technology in 2008. He is in charge of development and daily operation of the DoD Wireless Networking Library. His current research interests are in modeling, optimizing, and developing networking protocols for tactical edge networks including resilient transport, secure and resilient multimedia application, network-coding-based routing, and tactical MANET networking in general.

**Tommaso Melodia** is an Associate Professor with the Department of Electrical Engineering at the State University of New York (SUNY) at Buffalo, where he directs the Wireless Networks and Embedded Systems Laboratory. He received his Ph.D. in electrical and computer engineering from the Georgia Institute of Technology in 2007. He had previously received his "Laurea" (integrated B.S. and M.S.) and Doctorate degrees in telecommunications engineering from the University of Rome "La Sapienza", Rome, Italy, in 2001 and 2005, respectively. He is a recipient of the National Science Foundation CAREER Award, and coauthored a paper that was recognized as the Fast Breaking Paper in the field of Computer Science for February 2009 by Thomson ISI Essential Science Indicators and a paper that received an Elsevier Top Cited Paper Award. He is the Technical Program Committee Vice Chair for IEEE Globecom 2013 and the Technical Program Committee Vice Chair for Information Systems for IEEE INFOCOM 2013, and serves in the Editorial Boards of IEEE Transactions on Mobile Computing, IEEE Transactions on Wireless Communications, and Computer Networks (Elsevier), among others. His current research interests are in modeling, optimization, and experimental evaluation of wireless networks, with applications to cognitive and cooperative networking, ultrasonic intrabody area networks, multimedia sensor networks, and underwater networks.