A Tutorial on Encoding and Wireless Transmission of Compressively Sampled Videos

Scott Pudlewski and Tommaso Melodia

***Abstract*—Compressed sensing (CS) has emerged as a promising technique to jointly sense and compress sparse signals. One of the most promising applications of CS is compressive imaging. Leveraging the fact that images can be represented as approximately sparse signals in a transformed domain, images can be compressed and sampled simultaneously using low-complexity linear operations. Recently, these techniques have been extended beyond imaging to encode video. Much of the compression in traditional video encoding comes from using motion vectors to take advantage of the temporal correlation between adjacent frames. However, calculating motion vectors is a processing-intensive operation that causes significant power consumption. Therefore, any technique appropriate for resource constrained video sensors must exploit temporal correlation through low-complexity operations.**

In this tutorial, we first briefly discuss challenges involved in the transmission of video over a wireless multimedia sensor network (WMSN). We then discuss the different techniques available for applying CS encoding first to images, and then to videos for error-resilient transmission in lossy channels. Existing solutions are examined, and compared in terms of applicability to wireless multimedia sensor networks (WMSNs). Finally, open issues are discussed and future research trends are outlined.

***Index Terms*—Compressed Sensing, Multimedia communication, Wireless sensor networks, Video coding, Energy-rate-distortion.**

I. Introduction

ADVANCES in sensing, computation, storage, and wireless networking are driving an increasing interest in multimedia sensing applications [1], [2], [3], [4]. Specifically, video surveillance applications, where video sensors are used to implement a low cost, quickly deployable video surveillance system; and *participatory* [5], [6] sensing applications, which allow users of mobile devices to capture, disseminate and view information freely within a network, are two technologies that would not be feasible without low-cost, battery-powered, high-quality mobile video sensors. While these applications 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. However, 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], [8], [9], H.264/SVC [10])[1] through a layered wireless communication protocol stack, is not appropriate for wireless multimedia sensor networks (WMSNs) because of the following limitations:

- **Predictive Video Encoding is Computationally Intensive**. State-of-the-art predictive encoding requires calculating motion vectors, which is a computationally intensive operation. This requires significant power consumption and complexity at the sensor node. Since data transmission is also an energy-intensive task, compression is an essential component of any video transmission system. However, a WMSN system should ideally transfer most of the computational complexity to the multimedia sink, which is in general *not* a resource-constrained system [3].
- **Predictive Encoding of Video Increases the Impact of Channel Errors**. In existing layered protocol stacks (e.g., IEEE 802.11 and 802.15.4) frames are split into multiple packets. Any errors in even one of these packets, after a cyclic redundancy check, can cause visible distortion in a video frame. Because of the predictive nature of modern video encoders, distortion can then propagate to tens or even hundreds of frames that are dependent on the distorted frame. *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.*

Both of these limitations can lead to increased power consumption at the sensor nodes [12], [13]. Given the system hardware, increased computational complexity leads to increased power consumption. By reducing the computational complexity, we can reduce the power required to encode video. However, most common methods to reduce the complexity will generally lead to an *increase* in the compressed size of the encoded video, i.e., to a lower rate distortion performance. If the energy required to compress each video frame decreases, but the total amount of data to be transmitted increases, such an approach could actually increase the total energy required to transmit the video.

When dealing with channel errors, increasing the received signal to noise ratio (SNR) is often necessary to reduce the number of errors to an acceptable level. Since power is limited in real systems, other methods have been developed to decrease the BER. Traditionally, forward error correction (FEC) (e.g., Reed-Solomon [14] codes or RCPC [15] codes)

Manuscript received 7 October-2011; revised 10 April 2012 and 10 August 2012. This paper is based upon work supported by the Office of Naval Research under grant N00014-11-1-0848 and by the National Science Foundation under grant CNS1117121.

S. Pudlewski was with Department of Electrical Engineering, State University of New York (SUNY) at Buffalo and is now with Lincoln Laboratory, Massachusetts Institute of Technology (e-mail: scott.pudlewski@ll.mit.edu).

T. Melodia is with Department of Electrical Engineering, State University of New York (SUNY) at Buffalo (e-mail: tmelodia@eng.buffalo.edu).

Digital Object Identifier 10.1109/SURV.2012.121912.00154

[1]For more detail as to the implementation of these encoders, the reader is referred to [11].

is employed to reduce the BER for a fixed SNR. However, FEC will increase the size of each encoded packet, which could result in a net *increase* in total energy required for transmission. Automatic repeat-request (ARQ) is another method for dealing with bit errors, in which packets containing errors are retransmitted. This requires an increase in the number of packets transmitted, which again may increase the energy consumption.

Compressed sensing (CS) [16], [17], [18], [19], [20], [21], [22], is a promising technique for dealing with these limitations. Compressed sensing (aka "compressive sampling") is a new paradigm that allows the faithful recovery of signals from $M << N$ measurements where N is the number of samples required for the Nyquist sampling. Since these M measurements are created by taking M *linear combinations* of the N pixels, 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* [23], [24].

CS may provide an alternative to traditional video encoding techniques by combating both their *computational* and the *error resilience* limitations simultaneously. Traditionally, in WMSN platform designs [25], [26], [27], [28] these two limitations are viewed as a competing for a fixed energy budget (e.g., energy per video frame). Channel errors are compensated for by increasing the transmission power. However, this will decrease the amount of energy available for encoding the video. If we instead look at reducing the encoder complexity (for example switching from H.264 to MJPEG), the rate distortion performance generally decreases, causing the total amount of transmitted data to increase. For a fixed energy budget, this will decrease the energy available to transmit each bit, decreasing the SNR and increasing the susceptibility of the video to channel errors. CS encoding has the potential to reduce both the energy required to encode the image (because of the very low complexity) and the energy required to transmit the image (by reducing the SNR required to correctly decode the video at the receiver) *simultaneously*.

We will review the basic concepts of compressive imaging [29], [30], [31], [32] in Section IV. While these techniques can clearly take advantage of the spatial correlation within each video frame, these methods do not deal with the temporal correlation. In traditional video encoding, this temporal correlation is the main source of compression [8], but it is also the main cause of complexity. In Section V, we examine different ways to use CS encoded frames to avoid the need for motion vectors. While beyond the scope of this article, it is worth noting that another approach to compressive video sensing involves modifying the reconstruction process to take advantage of additional sparsity [33]. While interesting and potentially very effective, this article will focus on the CS based encoder.

The rest of this paper is structured as follows. Section II presents the challenges of video encoding in sensor networks. Compressed sensing is briefly introduced in Section III. Section IV, gives an introduction of compressive imaging, while Section V introduces the current state of the art in CS video encoding. In Section VI we discuss future trends in CS video encoding, and in Section VII we draw the main conclusions.

II. Challenges

Multimedia networking applications are normally characterized by high complexity and high data rate. However, sensor nodes are ideally low-cost, *low-complexity* battery operated devices that have a long network lifetime, which generally leads to a *lower data rate* than other types of networks. For a practical WMSN implementation, a video encoding system must be designed that can fit within these constraints. Below we examine some of the key constraints.

A. Data Rate Constraints

While there exist standardized medium access control (MAC) protocols that are able to provide a high enough data rate to wirelessly transmit multimedia content (e.g., 802.11 [34], WiMAX [35]), and there exist standardized protocols that are able to reduce the power consumption at each node to acceptable levels (Zigbee [36], Bluetooth [37]), achieving *both* at the same time is much more difficult. The standard energy saving technique in sensor network MAC protocols is for nodes to enter a sleep mode when they are not transmitting or receiving data. However, these sleep cycles reduce the achievable data rate, which may be unacceptable when multimedia traffic is being transmitted. For example, the usable data rate for 802.15.4 [38] is generally below $70\,\mathrm{kbit/s}$ [39]. However, even QCIF (176×144 pixels/frame) video could need more than twice that rate to achieve acceptable quality. Clearly, we must move beyond traditional sensor networking protocols. However, unlike traditional 802.11, we must still be aware of energy consumption in the system design.

B. Complexity Constraints

While high-end mobile devices have recently become commercially available (i.e., smartphones, tablets), WMSN sensor nodes should ideally be simple, low-complexity devices. These devices are much cheaper and have a much longer battery life than even low-end smartphones. However, this increase in battery life comes at the cost of a decrease in computational capabilities. Nearly all of these energy efficient scalar sensors only contain 8- or 16-bit processors with very limited RAM memory, and are unable to implement complex video encoding algorithms.

It has been shown [40], [41], [42] for specific processors that, for traditional image and video encoding algorithms, 32-bit processors may be more energy efficient then 8- or 16-bit processors. This is because, while each 32-bit operation will consume more energy, the algorithms require fewer operations overall resulting in lower overall energy consumption. While this would probably hold true for the CS-based algorithms presented here, this is difficult to discuss in general because the efficiency of linear algebra operations is strongly dependent on both the processor hardware and the software implementation of the algorithms [43], [44]. The major advantage of CS-based algorithms is that, since the algorithm is itself very simple, it does not *require* a 32-bit processor to implement a real time streaming video system, and could be implemented on commercially available *scalar* sensor network devices.

Some attempts have been made to implement video on low complexity devices using traditional encoding methods. The

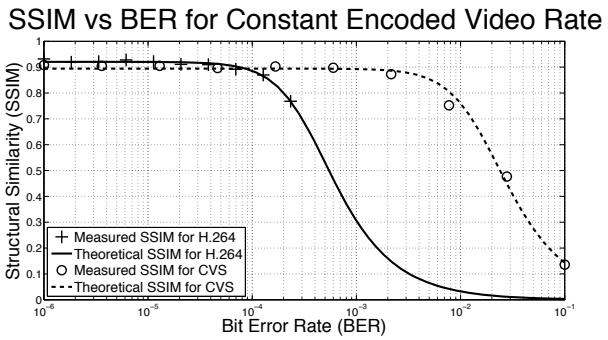

Fig. 1. SSIM [48] vs BER for H.264 and CVS Encoders

best known example of this is motion JPEG [45] (MJPEG), where a video is encoded as a series of JPEG encoded images. While MJPEG has become popular in devices such as digital cameras and cellphones, it is clearly not an ideal solution. For example, MJPEG does not take advantage of temporal correlation in a video sequence. In addition, JPEG image encoding still requires the source node to capture and temporarily store an entire raw video frame and perform a DCT [46] transform on each block of the image. While far less complex than motion vector calculations, these are still not insignificant operations. In Section IV, we will show how CS theory allows us to create an imaging system that requires very little complexity (hardware or software), while still taking advantage of temporal correlation to encode high-quality video.

C. Channel Constraints

A major challenge in WMSNs is compensating for lossy channels. We will discuss two aspects of wireless transmission that complicate the transmission of video, namely bit errors and multipath fading.

Bit Errors: It is well known that predictively encoded video is very susceptible to bit errors. In data networks, bit errors are usually dealt with using some form or ARQ or FEC. Both of these methods generally have an all-or-nothing approach to error correction, in that a received packet is either *entirely correct* or is *discarded and must be retransmitted*. However, even though video is less tolerant to bit errors than images, it is much more tolerant of bit errors than data networks due to error concealment techniques [32] [47]. While the quality does decrease sharply when the BER increases beyond some threshold, for low levels of BER there is *no measurable decrease in video quality*. This is shown in Fig. 1 for traditional H.264 encoding and a compressive video sensing (CVS) encoder (which will be described in detail in Section V-C).

This leads to an obvious tradeoff between the quality of the received video and the techniques used to reduce the BER. As is shown in Fig. 1, there is little or no effect in the perceivable quality in the received video for BER rates of up to 10^{-4} for H.264 or for 10^{-3} for CVS. One advantage of CS encoded images and video is that, because of independence between samples within an image, many more errors can be tolerated before significant quality degradation is noted in the received video.

Fading: While bit errors alone can cause major problems for video transmission if not accounted for, fading can also cause video quality to decrease significantly. One of the major problems associated with a fading channel is the correlation of errors in time. The bit errors will tend to be grouped together within a single packet, rather than spread out randomly among the entire transmission. This can cause problems when using FEC to correct errors. When bit errors are grouped together within a single packet, there may be too many errors in that one packet for the FEC code to correct, leading to the loss of that packet.

This is currently dealt with using schemes such as data interleaving [49], where the data is reordered in a non-contiguous way. When the data is de-interleaved, the grouped errors are effectively spread out. As long as the data is spread out "enough", this will relieve the problem. However, when data is interleaved, the receiver must wait until all non-contiguous portions of the data are received before it can reconstruct the data, causing an increase in latency. Similar to the BER discussion above, if errored video samples could be dropped without hindering the decoding of the correctly received samples, the "error grouping" effect of a fading channel would have no negative impact on received video performance, without the need for interleaving video samples.

D. Cost Constraints

Finally, to be feasible in a large scale, WMSN nodes should be as inexpensive as possible. While a cost analysis is beyond the scope of this paper, we mention this because, while more expensive processors and bigger batteries may solve many of the challenges posed above, this is not a realistic solution for WMSNs [3], [1]. For a rough estimate, we would like to keep the cost of the WMSN node to around the cost of a comparable scalar sensor node not taking the actual camera into account, i.e., around $50 USD.

III. Compressed Sensing Basics

In this section we introduce the basic concepts of compressed sensing as applied to image compression. We consider an image signal represented through a vector $\mathbf{x} \in \mathbb{R}^N$, where N is the number of pixels in the image and each element of the vector 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 [50]. The signal is measured by taking $M < N$ samples of the element vectors through a linear measurement operator Φ, defined by

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

We would like to recover $\mathbf{x}$ from measurements in $\mathbf{y}$. However, since $M < N$ the system is underdetermined. Hence, 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

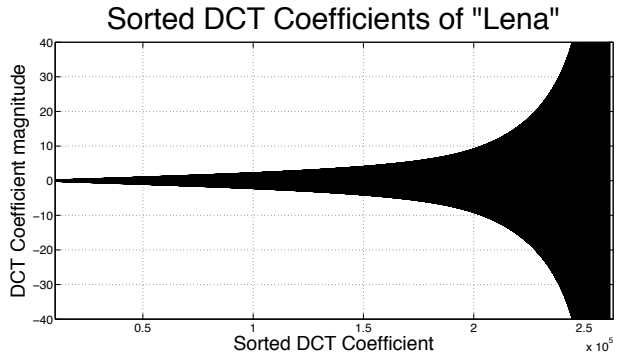

Fig. 2. DCT coefficients of Lena sorted in ascending order.

null space of $\tilde{\Psi}$), is also a solution of (2). However, it was proven in [18] that if the measurement matrix Φ is sufficiently incoherent with respect to the sparsifying matrix Ψ, 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 the optimization problem

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

which finds 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 [51]. 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:

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

where ϵ is a small tolerance.

Formally, any sampling matrix Φ must satisfy the uniform uncertainty principle (UUP) [18] [29]. The UUP states that if enough samples are taken, such that

$$M \geq KlogN, \tag{5}$$

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 intuitively see the association between UUP and sparse reconstruction [29], suppose that (6) holds for sets of size $2K$. If our K-sparse vector $\mathbf{y}$ is measured as $\mathbf{y} = \Phi\mathbf{s}_0$, then there can not be any other K-sparse or sparser vector $\hat{\mathbf{s}}' \neq \mathbf{s}_0$ that leads to the same measurements. If there were such a vector, then the difference $\mathbf{h} = \hat{\mathbf{s}}_0 - \hat{\mathbf{s}}'$ would be $2K$-sparse and have $\Phi\mathbf{h} = 0$. However, this is not compatible with (6).

Practically, this tells us that if for some N pixel image we choose M such that $M \geq K\ logN$, then we can reconstruct the K largest sparse components of the original image [18]. This allows us to choose the number of linear pixel combinations to achieve a specific quality based on the image we would like to compress. We will show in detail how these components are chosen, along with how the sampling matrix is implemented, in Section IV.

Note that (4) is a convex optimization problem [52]. The reconstruction complexity equals $O(M^2N^{3/2})$ if the problem is solved using interior point methods [53]. Although more efficient reconstruction techniques exist [54], 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.

IV. Compressive Imaging

Before discussing CS video, we will first introduce CS imaging. Compressive imaging is the basis for all of the video streaming systems that will be discussed later in Section V.

A. Compressive Imaging Background

It is clear that since most images can be represented in a sparse domain (e.g., wavelet or DCT), they can be sampled and compressed using (2) and recovered using (4). In this section we will examine some of the properties of images that have been compressed using this CS system, and how these properties can help address the challenges described in Section II.

Effects of Approximate Sparsity: In Section III, 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 [55] sorted in increasing order. While the image is clearly compressible, few if any of the DCT coefficients are *exactly* 0.

When we use (4) to reconstruct Lena with $M < N$, the reconstruction process will force the smaller coefficients to be exactly 0 [19], 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.

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 [55] encoded using (2). These tests were done using the wavelet transform as the sparsifying transform and reconstructed using the gradient projection for sparse reconstruction GPSR [56] algorithm. As M is increased and more samples are used in the image reconstruction, the SSIM of the image approaches 1.

Image distortion can be modeled [47] [2] as

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

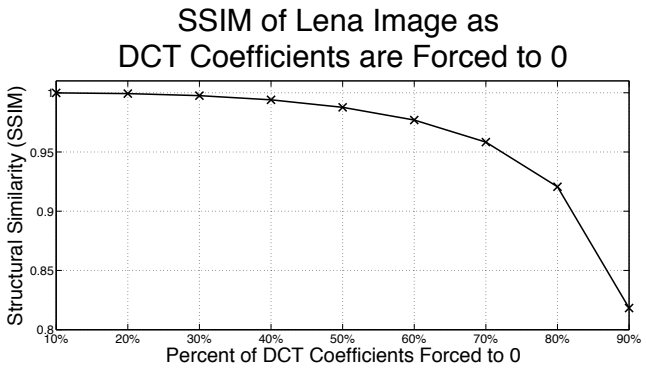

Fig. 3. SSIM of Lena after DCT transform, forcing the smallest coefficients to zero and inverse DCT transform.

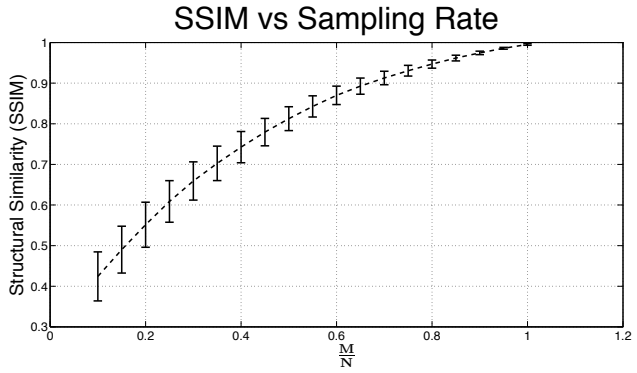

Fig. 4. SSIM vs sampling rate $\frac{M}{N}$.

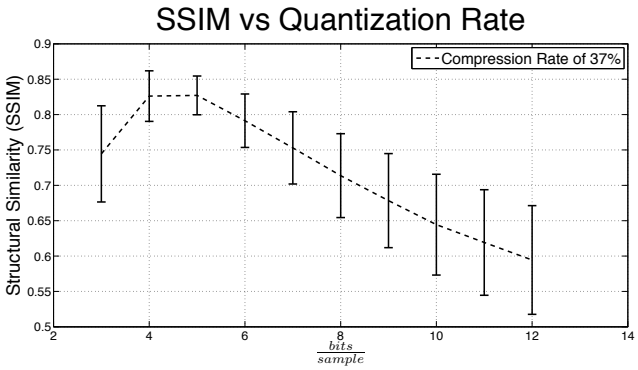

Fig. 5. SSIM vs quantization bits.

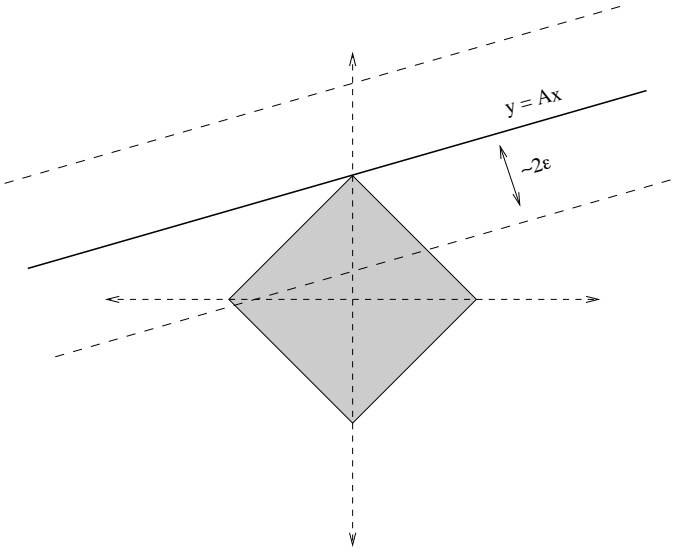

Fig. 6. Geometric interpretation of ℓ_1 norm minimization.

where D_0, Θ 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. 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.

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 [57]. 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 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 [19], 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 Φ 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[2]. 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 2ϵ 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 ℓ_1 norm ball. The two dashed lines represent the maximum variation in the samples when corrupted by additive noise of magnitude ϵ. 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 system represented in Fig. 6, the magnitude of ϵ would have to be about $\frac{1}{3}$ of the signal power before an incorrect "corner" of the norm ball is selected.

[2]For practical systems, C is a small constant between 5 and 10 [19].

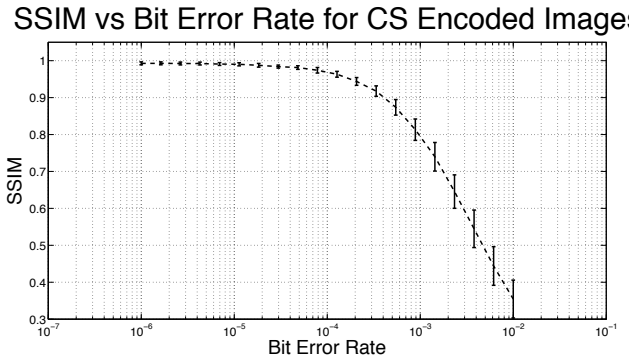

Fig. 7. Compressed Sensed Images Reconstructed With Bit Errors

Effects of Bit Errors: Though (7) accurately models the video quality when there are no errors, any bit errors may add further distortion to the received image. As shown in Fig. 7, the video does *not* have to be received perfectly for it to be acceptable at the receiver. At low BER rates, there is almost no effect in the received SSIM. As the BER increases past a certain level, however, the video quality drops off significantly.

Based on this observation, we have modeled the error performance as a low pass filter [12], [13] using

$$U(r_v) = \frac{\alpha(r_v)}{\sqrt{1+\tau^2(BER(r_v))^2}} \tag{10}$$

where $r_v = \beta \cdot \gamma$ is the encoded video rate in $\mathrm{kbit/s}$ as a function of the sampling rate and $U(r_v)$ is the quality of the received video in SSIM as a function of r_v. $\alpha(r_v)$ is the quality of the signal based only on the compression, and is calculated as in (7). The encoder-dependent constant τ is used to indicate where the quality begins to decrease. For a constant power budget *per image*, $BER(r_v)$ is clearly a function of r_v, since as more bits are needed to represent an image, each bit will be transmitted at lower power to keep the total power budget constant, reducing the SNR and therefore increasing the BER.

This gives us an estimate of the received video quality as a function of both the encoding rate and the channel conditions. While empirical in nature, this function has been shown to accurately predict the quality of a number of typical video encoders [12], [13], and can be used to compare the received quality of a video or image in specific channel conditions for a given encoding rate. We present this function here to demonstrate that there is a tradeoff between the distortion caused at the encoder and the distortion caused by channel errors. As more samples are used to encode a video frame, the quality of a perfectly received video will be increased (i.e., Fig. 4). But if less power is used for each sample, the SNR of the transmitted image will be higher, leading to an increased BER, and therefore a decrease in the received quality (i.e., Fig. 7).

Sampling Complexity: Traditional image compression schemes generally 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 8×8 pixel blocks. Then each of these 64 pixel groups are transformed using a DCT transform. JPEG2000 [58] 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 [59] 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 Φ. Assume we are dealing with a 512×512 pixel image, and that M is set at $\frac{N}{5}$. This will result in a Φ matrix that is $52,429 \times 262,144$. A direct implementation would require 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 [60], defined as

$$Y = H_{32} \cdot X, \tag{11}$$

where Y represents a matrix of image samples (measurements), H_{32} is the 32×32 Hadamard matrix and X is a matrix of the image pixels that 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 $\mathbf{x}$ 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).

Encoder Energy Cost: There are two components to take into consideration when evaluating the energy consumption of a wireless video sensor. First is the encoder complexity and how that affects the energy required to encode the video. Second is the energy required to *transmit* the video over a wireless link or network. A comparison of the CVS encoder to traditional video encoding (i.e., H.264 [7] and MJPEG [45]) was carried out in detail in [12], [13].

To summarize those results, the *energy constrained* SNR for energy budget E_B is defined as

$$SNR(r_{ch}, r_v)_{E_B} = \frac{L \cdot r_{ch} \cdot d_{free} \cdot (E_B - E_E(r_v))}{N_0 \frac{r_v}{r_{ch} \cdot fps}}, \tag{12}$$

where L is the path loss, N_0 is the noise power, d_{free} is the free distance of the channel code, r_{ch} is the channel coding rate, r_v is the video encoding rate, and fps is the framerate of the video in frames per second. The encoding energy E_E, which is dependent on r_v, is determined empirically for a encoder-platform combination. The important thing to note is that as r_v increases, the energy needed to encode the video increases while the transmission energy per bit decreases, causing the SNR to decrease.

Using this model, the performance of CVS was compared to H.264 and MJPEG for two common platforms. The results of this are presented in Fig. 8, which represents video encoded on a relatively high powered node (i.e., a notebook computer), and Fig. 9, which represents video encoded on a mobile smartphone. In both of these simulations, there is a tradeoff between energy and received video quality. The CVS encoder

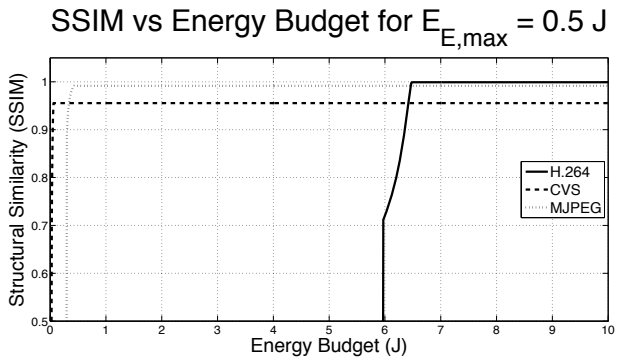

Fig. 8. SSIM vs. Total Energy Budget for $E_{E,max} = 0.5J$.

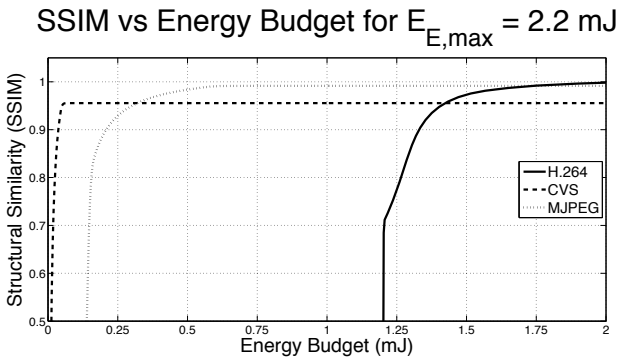

Fig. 9. SSIM vs Total Energy Budget for $E_{E,max} = 2.2mJ$

results in a lower maximum received video quality (from a strict rate-distortion perspective), but can generally achieve that quality at a much lower energy requirement than either MJPEG or H.264. 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. 9. 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 encoder crosses at 1.25 mJ. This means that *we can achieve the same quality for much lower energy cost using CVS*.

B. Single Pixel Camera

A major step in making the theoretical CS imaging applications practical was the development of the single pixel camera [30]. The single pixel camera is able to simultaneously measure and compress images using very simple hardware. The camera uses a Texas Instruments digital micromirror device (DMD) [61] to reflect a scene onto a single photodiode. The DMD is able to individually change the angle of each mirror from a bank of 1027×768 mirrors to either $+12^\circ$ or -12° from horizontal. Using a biconvex lense, this allows the system to aim a subset of the mirrors at the photodiode. The output of this photodiode is then amplified and quantized to produce a single CS sample. This is then repeated to produce M samples. Each of these samples is then passed through an analog to digital converter, and either transmitted or stored in memory.

The importance of this is twofold. First, the image capture and compression is done simultaneously, allowing the entire camera-encoder system to consist of a single DMD and an analog-to-digital converter. All of the signal processing is done implicitly when the intensity at the photo-detector is measured. Another less obvious advantage is that, since only a single photodiode is used, an infrared imaging system can be built without increasing the cost significantly over the visual light system.

C. Advantages of CS over JPEG

There are two very important advantages that CS imaging has over JPEG imaging. First, CS imaging can compress an image with far lower computational complexity. While it is difficult to get an accurate measurement between the two, note that, neglecting the DMD, the entire CS imaging described in Section IV-B is lower by a factor of $\frac{M}{N}$ than the complexity required *simply to capture the image in a JPEG based device*.

The other advantage is the performance of CS imaging in a noisy channel. 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 schemes 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. 10, where the set of images [55] are encoded using CS and transmitted over a lossy channel[3]. 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. 10. 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. Figure 11 shows the average SSIM of the SIPI images [55] transmitted through a binary symmetric channel with varying BER. The quality of CS-

[3]For implementation, the image is encoded using CS, quantized, given a four byte header containing the basic encoding parameters and stored in a file or transmit buffer. It is then packetized and transmitted. The encoding matrix is assumed to be predefined and known throughout the network.

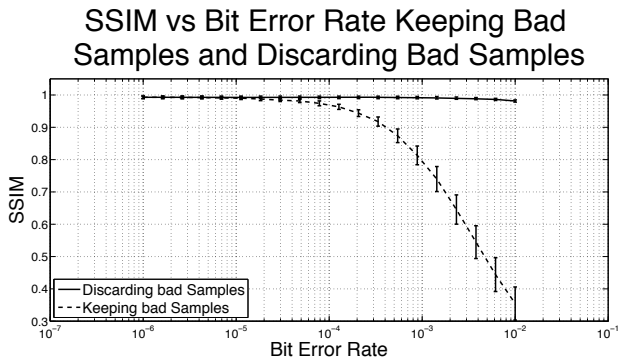

Fig. 10. Compressed Sensed Images Reconstructed With and Without Incorrect Samples

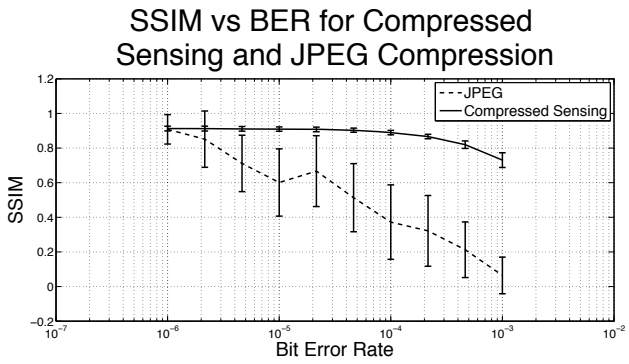

Fig. 11. Structural Similarity (SSIM) vs Bit Error Rate (BER) for compressed sensed images, and images compressed using JPEG

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. 12, 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}.

V. Compressive Video

We have seen how CS can be used to encode an image for transmission over WMSN. In the following, we describe how these concepts can be extended to video encoding.

A. Video as a Series of Images

As observed above, CS image encoding systems are generally much less computationally complex than traditional image and video encoding processes. Based on this, the most straightforward video encoding scheme is to take each frame of a video individually, treat it as an image and use CS image encoding schemes. Although this approach seems very simplistic, it is conceptually analogous to the very common MJPEG. As the most straightforward video encoder, we will start here.

Formally, if a series of video frames can be defined by $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_L}$ for L video frames, then we define the encoded video as a series of compressed frames $\mathbf{y_1}, \mathbf{y_2}, \ldots, \mathbf{y_L}$ where $\mathbf{y_i} = \Phi_i \mathbf{x_i}$. At the receiver, each frame $\hat{\mathbf{x}}_\mathbf{i}$ is reconstructed by solving (4) with $\mathbf{y} = \mathbf{y_i}$ and $\Phi = \Phi_i$.

While such a system would indeed have low complexity at the sensor nodes, the *temporal correlation* between consecutive frames is ignored. There are however methods for taking advantage of this correlation without using motion vectors. Below, we describe the state-of-the-art in video encoders using compressed sensing.

B. Hybrid Video Compression Schemes

Recent work was done attempting to merge 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 the video.

Distributed Compressed Video Sensing (DISCOS): In [62], the authors present Distributed Compressed Video Sensing (DISCOS). 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 encoded using CS through a combination of both frame-based calculations (linear combinations of the entire frame) and block-based calculations (linear combinations of a set of pixels restricted to a set of non-overlapping blocks).

The core difference between DISCOS and traditional video encoding is that, rather than traditional sparsifying operations (i.e. wavelet, DCT), each CS block is represented using a matrix composed of a dictionary of *temporal neighboring blocks*. While this is shown in [62] to far outperform other methods, it does require a "motion-vector-like" operation at the source node. Because of the high complexity of motion vector calculations, they cannot be implemented on a simple low-complexity and energy constrained sensor node, and are therefore not appropriate for WMSNs.

Block-Based Compressive Video Sampling: The authors of [63] also present 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.

While this encoder does not leverage temporal correlation specifically, the use of a key frame to *estimate* the sparsity of subsequent frames is a novel concept that is shown to lead to very good video rate-distortion performance. However, as with the previous system, this system uses traditional motion vector techniques and is therefore not suitable for WMSN implementations.

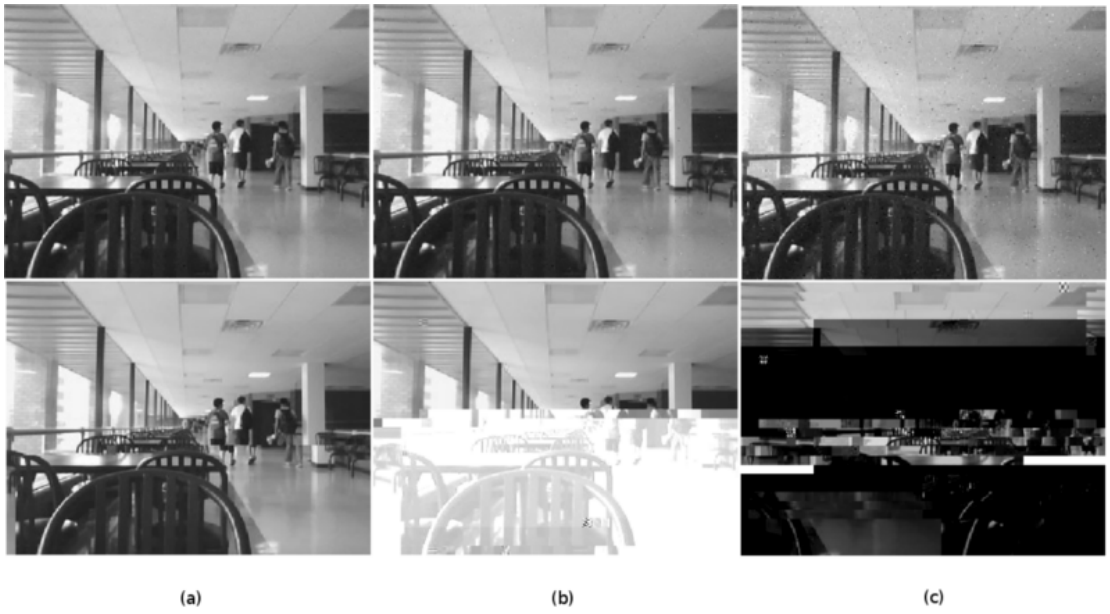

Fig. 12. Image compressed with CS (above) and JPEG (below) for BER (a) 10^{-5} (b) 10^{-4} (c) 10^{-3}.

C. Compressive Video Sensing (CVS)

We have developed a compressive video sensing (CVS) video encoder [64]. CVS leverages the linear nature of CS encoding to take advantage of temporal correlation between CS encoded video frames. The CVS video encoding process is divided into intra-frames (I frame) and inter-frames, or progressive frames (P frames). The video encoding process is illustrated in Fig. 13. For a given sampling rate and physical input (i.e., image), the video controller will generate either an I or a P frame, both of which are described below. The pattern of the encoded frames is $IPPP\cdots PIPPP\cdots$, where the distance between two I frames is referred to as the group of pictures (GOP).

1) Intra-frame (I) Encoding: Each of the I frames are encoded individually, i.e., as a *single image* that is *independent of the surrounding frames*. One of the main goals of this encoder is to maximize the quality of a received video for given encoded video rate. This is done mainly by adjusting two parameters to the I frame encoder.

Sample Quantization Rate. Based on the tests discussed in Section IV and reported in Fig. 5, we can see that as Q decreases and less bits are being used to encode each sample, more samples can be obtained for the same compression rate. As long as the uncertainty (here measured as the magnitude of the quantization error) in each sample does not get too large, this process will result in a more accurate reconstruction.

Sampling Rate γ. The sampling rate γ, where $\gamma = \frac{M}{N}$ and $0 < \gamma \leq 1$, is the number of transmitted samples per original image pixel. An empirical study was performed on the SIPI image database [55] to determine the amount of distortion in the recreated images due to varying sampling rates, and is reported in Fig. 4.

2) Progressive (P) frame Encoding: The I frame encoding is, like the basis for the other encoders presented here, based on CS encoding of each image independently. To take advantage of temporal correlation, we consider the *algebraic difference between the CS samples*. The motivation behind this is that a CS encoded image is simply a series of linear combinations of subsets of the pixels of an image, which is represented by the multiplication by the sampling matrix Φ. Now assume that we have two frames $\mathbf{x_i}$ and $\mathbf{x_{i+1}}$. For most CS applications, it is assumed that the transmitting sensor node does not have access to the raw image data; in this case $\mathbf{x_i}$ and $\mathbf{x_{i+1}}$. Instead, the sensor node only has access to $\mathbf{y_i} = \Phi\mathbf{x_i}$ and $\mathbf{y_{i+1}} = \Phi\mathbf{x_{i+1}}$. However, as long as Φ is kept constant, it is easy to see that if we calculate a difference vector $\mathbf{dv}$ as

$$\mathbf{dv_{i+1}} = \mathbf{y_{i+1}} - \mathbf{y_i}, \tag{13}$$

then this is equivalent to

$$\begin{aligned} \mathbf{dv_{i+1}} &= \Phi\mathbf{x_{i+1}} - \Phi\mathbf{x_i} \\ &\Phi(\mathbf{x_{i+1}} - \mathbf{x_i}), \end{aligned} \tag{14}$$

which is the same as if we had sampled the difference between the two frames explicitly.

Then, each dv_{i+1} is *again compressively sampled* and transmitted. If the image being encoded x_{i+1} and the reference image x_i are very similar (i.e., have a very high correlation coefficient), then dv_{i+1} will be sparse (in the domain of compressed samples) and have less variance than either of the original images. The main compression of the difference frames comes from the above properties and is exploited in two ways. First, because of the sparsity in the difference frame, dv_{i+1} can be further compressed using CS. The number of samples needed is based on the sparsity as in the CS sampling of the initial frame. Second, the lower variance allows us to use fewer quantization levels to accurately represent the information, and therefore fewer bits per sample.

Intuitively, each I frame is temporarily stored at the sender. After the P frame samples are taken, the source subtracts these samples from the stored I frame samples. Because the

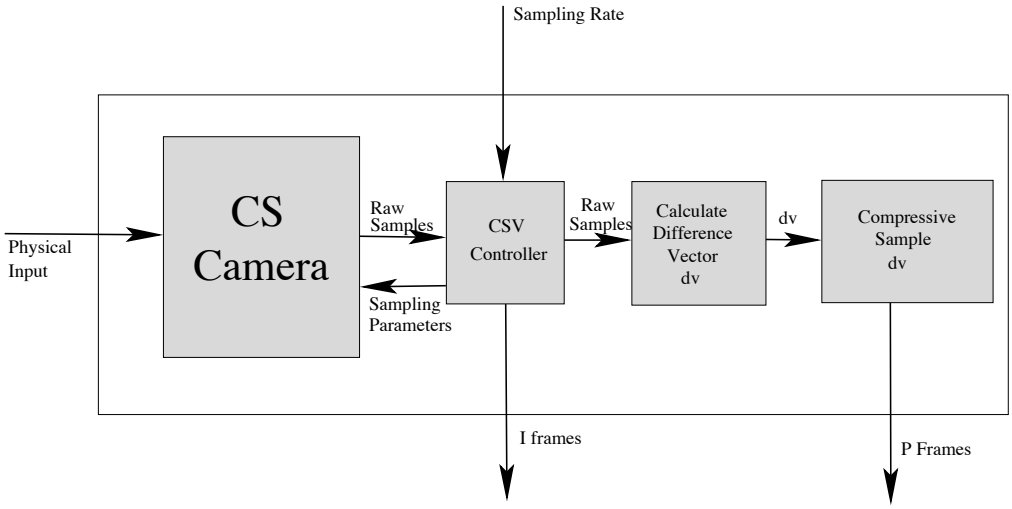

Fig. 13. Block diagram for CS video encoder.

TABLE I
COMPRESSION GAIN USING P FRAMES.

Amount of Motion	low	medium	high
Gain	556%	455%	172%

difference between two images will be sparse in the image domain, and the CS measurements are taken as a weighted summation of a small subset of pixels (as in (11)), the difference vector will itself be sparse. In practice this vector after quantization is very sparse[4]. Since the sparsity of the signal determines how well it can be compressed using CS techniques, CS is a natural choice for compression of the difference vector.

Formally, $\mathbf{dv}$ is compressed using (2), quantized and transmitted. The number of samples m needed to represent $\mathbf{dv}$ after it is compressed is proportional to the *sparsity* K of $\mathbf{dv}$ and defined as $m \approx K\ log(N)$ where N is the length of $\mathbf{dv}$. For videos with very high temporal correlation such as security videos, the $\mathbf{dv}$ will also have very low variance, allowing for a lower quantization rate Q. In the simulations reported in this paper, we used $Q = 3$.

In terms of compression ratio, the effectiveness of this scheme depends on the temporal correlation between frames of the video. The compression of each of these schemes (at the same received video quality) was compared to basic CS compression (i.e., using I frames only) for three videos. The videos chosen were Foreman (representing high motion) and two security videos; one monitoring a walkway with moderate traffic (moderate motion) and one monitoring a walkway with only light traffic (low motion), and the percentage improvement, calculated as $\frac{\text{Size without P frames}}{\text{Size with P frames}} \times 100$ is presented in Table I. While the compression of the high motion video can be increased by 172%, the moderate and low motion security videos (which represent typical application scenarios for our encoder) show far more improvement by using the P frames.

[4]In the experiments presented in this paper, more than 90% of the elements of the $\mathbf{dv}$ were 0 after quantization for the majority of the videos, and the worst case was 68%.

3) Video Decoding: The CVS video decoder is shown in Fig. 14. After decoding (i.e., demodulation and detection) the receiver determines whether the received sample is an I frame or a P frame. If the received frame is an I frame, then the decoder simply solves (4) based on the received samples. These samples are also stored for use in decoding the P frames, and are defined as $\hat{\mathbf{y}}_\mathbf{I}$. If the received frame is a P frame, then the received samples represent a difference vector $\hat{\mathbf{dv}}$. Once $\hat{\mathbf{dv}}$ is reconstructed, again by solving (4), the samples of the P frame are calculated from $\hat{\mathbf{y}}_\mathbf{P} = \hat{\mathbf{dv}} + \hat{\mathbf{y}}_\mathbf{I}$, and the P frame can be reconstructed.

D. Distributed Compressive Video Sensing

Recent work in distributed video coding (DVC) [65] has shown that much of the complexity of video encoding can be passed to the receiver. The basic concepts of DVC are very intuitive. Assume that, using entropy encoding, we can achieve an encoding rate of $R_X \geq H(X)$ on signal X, and $R_Y \geq H(Y)$ on signal Y, where $H(S)$ is the entropy of signal S [65]. Since we are able to tolerate some error in the recovered signal, we can establish a rate region of

$$\begin{aligned} R_X + R_Y &\geq H(X,Y) \\ R_X &\geq H(X|Y) \\ R_Y &\geq H(Y|X), \end{aligned} \tag{15}$$

which states that the sum of the rates of X and Y can achieve the joint entropy $H(X,Y)$. Surprisingly, this can be done using *separate encoding* (but *joint decoding*) of the two signals.

This is then extended in [66] to use CS encoded images. To accomplish this, 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), this correlation will be very high. This allows us to view frame S and 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. This allows us to create error correction bits

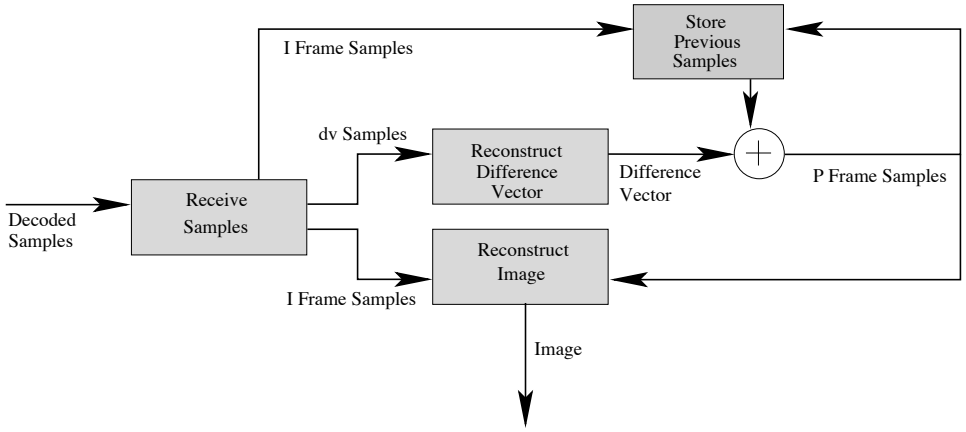

Fig. 14. Block diagram for CS video decoder.

for the second frame, and use these error correction bits to "correct" the differences between the two frames.

Formally, we assume that two consecutive frames $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ can be represented by

$$\begin{aligned} \mathbf{x}_i &= \mathbf{x}_{C_{i,i+1}} + \mathbf{x}_{U^i_{i,i+1}} \\ \mathbf{x}_{i+1} &= \mathbf{x}_{C_{i,i+1}} + \mathbf{x}_{U^{i+1}_{i,i+1}} \end{aligned} \tag{16}$$

where $\mathbf{x}_{C_{i,i+1}}$ is the portion of the frame common to both $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$, $\mathbf{x}_{U^i_{i,i+1}}$ is the portion of $\mathbf{x}_i$ that is unique with respect to $\mathbf{x}_{i+1}$ and $\mathbf{x}_{U^{i+1}_{i,i+1}}$ is the portion of $\mathbf{x}_{i+1}$ that is unique with respect to $\mathbf{x}_i$. Assuming $\mathbf{x}_i$ is a *key* or I frame, traditional video encoding would encode only $\mathbf{x}_i$, and $\mathbf{x}_{U^{i+1}_{i,i+1}}$, relying on the recovered version of $\mathbf{x}_i$ at the receiver for $\mathbf{x}_{C_{i,i+1}}$, which is needed to fully decode $\mathbf{x}_{i+1}$. However, for such a system to work, $\mathbf{x}_{C_{i,i+1}}$ must be determined at the encoder for each frame i, which is computationally very expensive.

However, if the destination can *estimate* the portion of the image that will be consistent between $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$, this information can be used at the receiver to reconstruct $\mathbf{x}_{i+1}$. The authors of [66], develop such a system where the side information of frame $\mathbf{x}_{i+1}$, denoted as $\mathbf{Si}_{i+1}$, can be estimated at the destination using a frame rate up-conversion tool [67]. Side information is then used as the starting criterion in a modified version of GPSR, which will reduce the number of iterations needed to reconstruct the non-key frames correctly.

The authors of [66] then propose a modification of the stopping criterion of the GPSR algorithm so the quality of the reconstructed video frame is kept sufficiently high without incurring excessive computation. The basis of this is to model the correlation between the j^{th} frame $\mathbf{x}_j$ and its side information $\mathbf{Si}_j$ at pixel p as a Laplacian distribution

$$P(\mathbf{x}_j(p) - \mathbf{Si}_j(p)) = \frac{\kappa(\mathbf{x}_j, \mathbf{Si}_j)}{2} e^{\kappa(\mathbf{x}_j, \mathbf{Si}_j)|\mathbf{x}_j(p) - \mathbf{Si}_j(p)|}, \tag{17}$$

where $\kappa(\mathbf{x}_j, \mathbf{Si}_j)$ is a model parameter defined by $\kappa(\mathbf{x}_j, \mathbf{Si}_j) = \frac{\sqrt{2}}{\sigma(\mathbf{x}_j, \mathbf{Si}_j)}$ and $\sigma(\mathbf{x}_j, \mathbf{Si}_j)$ is defined as the standard deviation of $\mathbf{x}_j(p) - \mathbf{Si}_j(p)$. The higher the correlation between $\mathbf{x}_j$ and $\mathbf{Si}_j$, the larger the parameter $\kappa(\mathbf{x}_j, \mathbf{Si}_j)$ will be. Since GPSR is an iterative reconstruction algorithm, we can then define the i^{th} iteration of the reconstructed version of $\mathbf{x}_j$ as $\tilde{\mathbf{x}_j}^{(i)}$. The first stopping criterion is then defined as

$$\frac{|\kappa(\tilde{\mathbf{x}_j}^{(i)}, \mathbf{Si}_j) - \kappa(\tilde{\mathbf{x}_j}^{(i-i)}, \mathbf{Si}_j)|}{\kappa(\tilde{\mathbf{x}_j}^{(i-i)}, \mathbf{Si}_j)} \leq T_\kappa \tag{18}$$

where T_κ is a threshold. The criterion (18) minimizes the difference between the reconstructed frame and the side information generated at the receiver. This is different from standard GPSR, which terminates when the norm of the minimum between the reconstructed signal and the gradient of the reconstructed signal is below a tolerance.

The next two stopping criteria defined are based on the sparsity of the reconstructed solution. The authors define which stopping criterion to use based on the value of $\frac{M}{N}$ used on each non-key frame. The authors show through simulation that the proposed algorithm is significantly faster that standard GPSR, and is able to reconstruct the test videos with up to 4 dB PSNR higher quality.

E. Block Based CS Video

The single pixel camera was introduced in Section IV-B for imaging applications. This imaging system is immediately applicable to video acquisition [68]. 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.

This at first seems to make the system more complicated. Traditionally, a CS reconstruction system requires a vector of samples from the same image, but each sample from the single pixel camera represents its own image "snapshot", and these images represent a near continuous stream of images in a video. However, in many cases, it can be assumed that the image changes slowly across a group of these snapshots. In this case, a set of measurements can be aggregated to represent a single video frame. The number of samples per frame is dependent on both the number of samples required to represent the video at a desired quality, the desired frame rate of the resulting video and the properties of the DMD [61]. For the DMD presented in [30], the switching time is 22 kHz. This speed allows the camera to capture about 733 samples for each frame of a 30 fps video, 1,833 samples for each frame of a 12 fps video or 22,000 for each frame of a 1 fps video.

The authors of [68] present two methods for reconstructing the frames into a video. First is a version of the scheme presented in Section V-A. 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. The sampling process is the same as the single pixel camera sampling we have discussed earlier. Each frame x_i is sampled using Φ to create a sample vector y_i. This is repeated for $i = 1, \ldots, L$, creating L sample vectors. These samples can then be reconstructed at the receiver *as if the samples were all taken together using a 3-D sampling matrix*. This will add some computational complexity at the receiver. However, since 3-D wavelet transforms are well known, and since the problem of minimizing the 1-norm of a matrix is convex, this can still be solved in polynomial time. The major advantage is that we are *directly* taking the spatial correlation into account at the receiver without any additional complexity at the source.

While such a scheme is very promising, there is one major problem. The complexity of the reconstruction process is highly nonlinear. As stated above, traditional interior point methods have a complexity of $O(M^2N^{3/2})$. 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. However, if there is an application that does not require real time reconstruction, this is a simple system that will perform well.

VI. Future Research Challenges

While the current research in the application of CS techniques is promising, there are still a few challenges that need to be solved before this technology can be realized in a realistic network. In this section, we will introduce some of these challenges.

A. Reconstruction Complexity

This paper has been focused on the sampling and encoding of video using compressed sensing. However, the biggest hurdle in CS reconstruction is the complexity required to reconstruct a video. Currently, the most common reconstruction algorithms are either least absolute shrinkage and selection operator (lasso) [69] or gradient projection for sparse reconstruction (GPSR) [56]. Others commonly seen are orthogonal matching pursuit (OMP) [70], stagewise orthogonal matching pursuit (StOMP) [71], basis pursuit denoising (BPDN) [72], and many others (see for example [73]). While some of these algorithms are very fast, none of them can reconstruct video *in real time*, i.e., at 30 frames per second (or even 12 frames per second).

There are a few techniques for accomplishing this that may be promising. First is reducing the dimensionality of the signal, and reconstructing it in blocks, as is done in [62], [63], [74] and as described in Section V-B. As stated above, since the complexity of even the fastest algorithm is (much) more than linear, reconstructing four $\frac{N}{2} \times \frac{N}{2}$ images will be faster than reconstructing one $N \times N$ image. However, the "amount" of sparsity in an image is related to the *size* of the image. As the image is divided into smaller and smaller sub-images, the number of samples needed to reconstruct that image increases for the same reconstruction quality. This limits the practical applications of this technique.

Another processing technique for reducing the complexity is to use properties of the images in the reconstruction. For example, in [72], the authors present a scheme for iteratively updating a CS solution based on a previous solution. Since natural images are smooth, the difference in the sparse transforms of each column vector of an image can itself be represented as a sparse vector. This sparse column difference vector is then used to update the reconstruction of the previous column. The authors show that this system is indeed faster that others available. However, it is still not fast enough for real time decoding.

B. Adaptive Sampling Matrices

A major issue in CS encoding of images is that, while the compression is good, it does not generally compare to deterministic video compression methods. While we have shown that the *power required* to compress and transmit a video using CS techniques may be much lower than traditional methods [12], [13], reducing the compressed size of the video would present more applications for this technology.

One way to do this is to adapt the sampling matrix to the image, and increase the sparsity at the source. For instance, the sampling matrix Φ and the sparse transform matrix Ψ in (2) can be specifically chosen to optimize the rate-distortion performance at each frame. While there are some rather obvious techniques to accomplish this (such as the hybrid schemes described in Section V-B), to be practical, the system must be able to adapt to the properties of the video *without first sampling the entire video*. The system must be able to work on a single pixel camera or similar device.

VII. Conclusions

We have presented an introduction to compressed sensing as applied to video encoding. The goal of this work was to make a case for why CS should be used in video encoding for low power WMSN nodes. Currently available state-of-the-art algorithms are not suitable for sensor networks, and CS solves many of the problems associated with traditional methods. We have presented the background necessary to begin approaching this problem. We have also described some of the leading algorithms developed for applying CS to video.

References

[1] I. Akyildiz, T. Melodia, and K. Chowdhury, "Wireless multimedia sensor networks: Applications and testbeds," *Proc. IEEE*, vol. 96, no. 10, pp. 1588–1605, October 2008.

[2] S. Pudlewski and T. Melodia, "A Distortion-minimizing Rate Controller for Wireless Multimedia Sensor Networks," *Computer Communications (Elsevier)*, vol. 33, no. 12, pp. 1380–1390, July 2010.

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

[4] S. Soro and W. Heinzelman, "A Survey of Visual Sensor Networks," *Advances in Multimedia*, vol. 2009, Article ID 640386, 2009.

[5] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava, "Participatory sensing," in *Proc. ACM Sensys World Sensor Web Workshop (WSW)*, Boulder, Colorado, 2006.

[6] 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 Computing*, vol. 12, no. 4, pp. 12 – 21, July/August 2008.

[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 Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, July 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, April 2004.

[10] T.Wiegand, G. J. Sullivan, J. Reichel, H. Schwarz, and M.Wien, "Joint Draft 11 of SVC Amendment," Doc. JVT-X201, July 2007.

[11] Y. Wang, J. Ostermann, and Y.-Q. Zhang, *Video Processing and Communications*. Upper Saddle River, New Jersey: Prentice Hall, 2002.

[12] 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 Conference (GLOBECOM)*, Houston, TX, December 2011.

[13] S. Pudlewski and T. Melodia, "Compressive Video Streaming: Design and Rate-Energy-Distortion Analysis," *IEEE Trans. Multimedia*, in press 2013.

[14] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *J. Society for Industrial and Applied Mathematics*, vol. 8, no. 2, pp. 300–304, June 1960.

[15] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Trans. Commun.*, vol. 36, no. 4, pp. 389–400, April 1988.

[16] E. J. Candï¿½s, "Compressive Sampling," in *Proc. Intl. Congress of Mathematicians*, Madrid, Spain, 2006.

[17] D. Donoho, "Compressed Sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, April 2006.

[18] 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, February 2006.

[19] E.J. Candes and J. Romberg and T. Tao, "Stable Signal Recovery from Incomplete and Inaccurate Measurements," *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, August 2006.

[20] E. Candes and T. Tao, "Near-optimal Signal Recovery from Random Projections and Universal Encoding Strategies?" *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, December 2006.

[21] K. Gao, S. N. Batalama, D. A. Pados, and B. W. Suter, "Compressed sensing using generalized polygon samplers," in *Proc. Asilomar Conf. on Signals, Systems and Computers*, Pacific Grove, CA, November 2010, pp. 1–5.

[22] K. Gao, S. N. Batalama, and D. A. Pados, "Compressive sampling with generalized polygons," *IEEE Trans. Signal Process.*, submitted November 2010.

[23] M. Wakin, J. Laska, M. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. Kelly, and R. Baraniuk, "An Architecture for Compressive Imaging," in *Proc. of IEEE Intl. Conf. on Image Processing (ICIP)*, October 2006, pp. 1273 –1276.

[24] D. Takhar, J. Laska, M. Wakin, M. Duarte, D. Baron, S. Sarvotham, K. Kelly, and R. Baraniuk, "A new compressive imaging camera architecture using optical-domain compression," in *in Proc. SPIE Conf. on Computational Imaging IV*, San Jose, CA, January 2006, pp. 43 – 52.

[25] M. AlNuaimi, F. Sallabi, and K. Shuaib, "A survey of wireless multimedia sensor networks challenges and solutions," in *Proc. IEEE Conf. on Innovations in Information Technology (IIT)*, April 2011, pp. 191 –196.

[26] A. Seema and M. Reisslein, "Towards Efficient Wireless Video Sensor Networks: A Survey of Existing Node Architectures and Proposal for A Flexi-WVSNP Design," *IEEE Commun. Surveys Tutorials*, vol. 13, no. 3, pp. 462 –486, 2011.

[27] C. Yeo and K. Ramchandran, "Robust Distributed Multiview Video Compression for Wireless Camera Networks," *IEEE Trans. Image Process.*, vol. 19, no. 4, pp. 995 –1008, April 2010.

[28] B. Tavli, K. Bicakci, R. Zilan, and J. Barcelo-Ordinas, "A survey of visual sensor network platforms," *Multimedia Tools and Applications*, pp. 1–38, 2012, 10.1007/s11042-011-0840-z. [Online]. Available: http://dx.doi.org/10.1007/s11042-011-0840-z

[29] J. Romberg, "Imaging via Compressive Sampling," *IEEE Signal Processing Mag.*, vol. 25, no. 2, pp. 14–20, March 2008.

[30] M. Duarte, M. Davenport, D. Takhar, J. Laska, T. Sun, K. Kelly, and R. Baraniuk, "Single-Pixel Imaging via Compressive Sampling," *IEEE Signal Processing Mag.*, vol. 25, no. 2, pp. 83–91, March 2008.

[31] E. Candes and M. Wakin, "An Introduction To Compressive Sampling," *IEEE Signal Processing Mag.*, vol. 25, no. 2, pp. 21 –30, March 2008.

[32] S. Pudlewski and T. Melodia, "On the Performance of Compressive Video Streaming for Wireless Multimedia Sensor Networks," in *Proc. IEEE Intl. Conf. on Communications (ICC)*, Cape Town, South Africa, May 2010.

[33] Y. Liu, M. Li, and D. A. Pados, "Motion-aware decoding of compressed-sensed video," *IEEE Trans. Circuits Syst. Video Technol.*, in press.

[34] "IEEE Std 802.11b-1999/Cor 1-2001," 2001.

[35] "IEEE Std 802.16-2004," 2004.

[36] "IEEE 802.15 WPAN Task Group 4 (TG4)," http://grouper.ieee.org/groups/802/15/pub/TG4.html.

[37] "Specification of the bluetooth system - version 1.1b, specification volume 1 & 2," Bluetooth SIG, February 2001.

[38] "Draft Standard for Low-Rate Personal Area Networks," IEEE 802.15.4/D17, October 2002.

[39] G. Lu, B. Krishnamachari, and C. Raghavendra, "Performance evaluation of the IEEE 802.15.4 MAC for low-rate low-power wireless networks," in *Proc. IEEE International Conference on Performance, Computing, and Communications (IPCCC)*, Phoenix, Arizona, April 2004, pp. 701 – 706.

[40] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "NeXt Generation / Dynamic Spectrum Access / Cognitive Radio Wireless Networks: A Survey," *Computer Networks (Elsevier)*, vol. 50, no. 13, pp. 2127–2159, September 2006.

[41] I. Downes, L. B. Rad, and H. Aghajan, "Development of a Mote for Wireless Image Sensor Networks," in *Proc. COGnitive systems with Interactive Sensors (COGIS)*, Paris, France, March 2006.

[42] D. McIntire, "Energy Benefits of 32-bit Microprocessor Wireless Sensing Systems," Sensoria Corporation White Paper.

[43] L. S. Blackford, J. Demmel, J. Dongarra, I. Duff, S. Hammarling, G. Henry, M. Heroux, L. Kaufman, A. Lumsdaine, A. Petitet, R. Pozo, K. Remington, and R. C. Whaley, "An Updated Set of Basic Linear Algebra Subprograms (BLAS)," *ACM Trans. Mathematical Software (TOMS)*, vol. 28, no. 2, pp. 135–151, June 2002.

[44] J. Dongarra, "Basic Linear Algebra Subprograms Technical Forum Standard," *International J. High Performance Applications and Supercomputing*, vol. 16, no. 1, pp. 1–111, 2002.

[45] "Digital Compression and Coding of Continuous-Tone Still Images - Requirements and Guidelines," ITU-T Recommendation T.81, 1992.

[46] N. Ahmed and T. Natarajan and K. R. Rao, "Discrete Cosine Transform," *IEEE Trans. Computers*, vol. C-23, no. 1, pp. 90–93, January 1974.

[47] K. Stuhlmuller, N. Farber, M.Link, and B. Girod, "Analysis of video transmission over lossy channels," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 1012–1032, June 2000.

[48] 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, April 2004.

[49] J. G. Proakis and M. Salehi, *Fundamentals of Communication Systems*. Upper Saddle River, New Jersey: Prentice Hall, 2005.

[50] A. Graps, "An Introduction to Wavelets," *IEEE Computational Science and Engineering*, vol. 2, no. 2, pp. 50–61, 1995.

[51] A. Bruckstein, D. Donoho, and M. Elad, "From Sparse Solutions of Systems of Equations to Sparse Modeling of Signals and Images," *SIAM Review*, vol. 51, no. 1, pp. 34–81, February 2007.

[52] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, March 2004.

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

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

[55] USC Signal and Image Processing Institute, http://sipi.usc.edu/database/index.html.

[56] 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. Sel. Topics Signal Process.*, vol. 1, no. 4, pp. 586–598, 2007.

[57] Yousuf Baig and Edmund M-K. Lai and J.P. Lewis, "Quantization Effects on Compressed Sensing Video," in *Proc. IEEE Intl. Conf. on Telecommunications (ITC)*, Doha, Qatar, April 2010.
[58] "JPEG2000 Requirements and Profiles," ISO/IEC JTC1/SC29/WG1 N1271, March 1999.
[59] 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. Proc. SPIE 2569, 1995, pp. 68–79.
[60] L. Gan, T. Do, and T. D. Tran, "Fast Compressive Imaging Using Scrambled Block Hadamard Ensemble," in *Proc. European Signal Processing Conference (EUSICPO)*, Lausanne, Switzerland, 2008.
[61] Valiux, http://www.vialux.de/.
[62] T. Do, Y. Chen, D. Nguyen, N. Nguyen, L. Gan, and T. Tran, "Distributed compressed video sensing," in *Proc. IEEE Intl. Conf. on Image Processing (ICIP)*, November 2009, pp. 1393 –1396.
[63] V. Stankovic, L. Stankovic, and S. Cheng, "Compressive Video Sampling," in *In Proc. European Signal Processing Conf. (EUSIPCO)*, Lausanne, Switzerland, August, pp. 2–6.
[64] S. Pudlewski, T. Melodia, and A. Prasanna, "Compressed-Sensing-Enabled Video Streaming for Wireless Multimedia Sensor Networks," *IEEE Trans. Mobile Computing*, vol. 11, no. 6, pp. 1060–1072, June 2011.
[65] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed Video Coding," *Proc. IEEE*, vol. 93, no. 1, pp. 71–83, January 2005.
[66] L. W. Kang and C. S. Lu, "Distributed Compressive Video Sensing," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Taipei, Taiwan, April 2009, pp. 1169 – 1172.
[67] AviSynth MSU frame rate conversion filter, http://compression.ru/video/frame_rate_conversion/index_en_msu.html.
[68] 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 Symposium (PCS)*, Beijing, China, April 2006.
[69] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," *J. Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267 – 288, 1996.
[70] J. Tropp, "Greed is good: algorithmic results for sparse approximation," *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2231 – 2242, October 2004.
[71] D. L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, "Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit," *Stanford Technical Report*, 2006.
[72] M. Salman Asif and J. Romberg, "Dynamic updating for ell_1 minimization," *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 2, pp. 421 – 434, April 2010.
[73] Rice DSP Compressive Sensing Resources, http://dsp.rice.edu/cs.
[74] A. Wani and N. Rahnavard, "Compressive Sampling for Energy Efficient and Loss Resilient Camera Sensor Networks," in *Proc. IEEE Conf. on Military Communication (MILCOM)*, Baltimore, MD, November 2011.

Scott Pudlewski [M'2007] (scott.pudlewski@ll.mit.edu) received his B.S. in Electrical Engineering from the Rochester Institute of Technology, Rochester, NY in 2008, and his M.S. and Ph.D degrees in Electrical Engineering from the University at Buffalo, The State University of New York (SUNY), Buffalo, NY in 2010 and 2012 respectively. He is currently a Technical Staff Member at the Massachusetts Institute of Technology (MIT) Lincoln Laboratory in Lexington, MA. His main research interests include video transmission and communications, networking in contested tactical networks, convex optimization, and wireless networks in general.

Tommaso Melodia [M'2007] Tommaso Melodia (tmelodia@buffalo.edu) is an Associate Professor with the Department of Electrical Engineering at the University at Buffalo, The State University of New York (SUNY). 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 2006, respectively. He coauthored a paper that was 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 an Associate Editor for IEEE Transactions on Mobile Computing, Computer Networks (Elsevier), IEEE Communications Surveys and Tutorials, and ACM/Springer Wireless Networks. 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. His current research interests are in modeling, optimization, and experimental evaluation of wireless networks, with applications to cognitive and cooperative networking, multimedia sensor networks, and underwater networking.