Resilient Image Sensor Networks in Lossy Channels Using Compressed Sensing

Scott Pudlewski, Arvind Prasanna, Tommaso Melodia
Wireless Networks and Embedded Systems Laboratory
Department of Electrical Engineering
State University of New York (SUNY) at Buffalo
e-mail: {smp25, ap92, tmelodia}@buffalo.edu

***Abstract*—Data loss in wireless communications greatly affects the reconstruction quality of a signal. In the case of images, data loss results in a reduction in quality of the received image. Conventionally, channel coding is performed at the encoder to enhance recovery of the signal by adding known redundancy. While channel coding is effective, it can be very computationally expensive. For this reason, a new mechanism of handling data losses in Wireless Multimedia Sensor Networks (WMSN) using Compressed Sensing (CS) is introduced in this paper. This system uses compressed sensing to detect and compensate for data loss within a wireless network. A combination of oversampling and an adaptive parity scheme are used to determine which CS samples contain bit errors, remove these samples and transmit additional samples to maintain a target image quality**

A study was done to test the combined use of adaptive parity and compressive oversampling to transmit and correctly recover image data in a lossy channel to maintain Quality of Information (QoI) of the resulting images. It is shown that by using the two components, an image can be correctly recovered even in a channel with very high loss rates of 10%. The AP portion of the system was also tested on a software defined radio testbed. It is shown that by transmitting images using a CS compression scheme with AP error detection, images can be successfully transmitted and received even in channels with very high bit error rates.

I. Introduction

Wireless Multimedia Sensor Networks (WMSN) [1] are self-organizing wireless systems of embedded devices deployed to retrieve, distributively process in real-time, store, correlate, and fuse multimedia streams originated from heterogeneous sources. Even though multimedia content can be transmitted successfully even with some losses, it is still important to ensure that the *quality of the received content* (i.e. Quality of Information (QoI)) is maintained at an acceptable level for the end user.

Wireless transmissions are notoriously prone to losses [5]. Two main causes of data loss are bit errors due to noisy channels and missing packets due to transmitter or receiver errors. To combat this, forward error correction (FEC) is often used to add known redundancy into the data stream and allow the receiving node to detect and correct a fixed number of bit errors. The two types of FEC coding commonly used for this purpose are block coding such as Reed-Solomon coding [14] [18], and convolutional codes [15] [16] [17]. Although FEC coding is effective, it can be very computationally expensive. The advantage of using FEC in sensor networks has been demonstrated in recent work. For example, in [6] the MicaZ platform is used to evaluate the performance of turbo codes. It is shown that it is less energy consuming to use turbo codes than to retransmit lost data. However, this is true only for low data rate applications. Automatic Repeat reQuest (ARQ) is another way of handling losses that is based on timeouts and retransmissions of missing/incorrect data. However, for real time data streams (i.e., video, VOIP), transmitting old packets may result in the media being recreated out of order at the receiver.

Another challenge in WMSNs is the need for compression. The amount of data needed for many applications (such as images) requires that redundant information be removed from the data stream before transmission, thereby reducing the amount of data transmitted. One negative effect of this is that the “importance” of each transmitted bit increases. In the case of multimedia transmission, the loss of a small amount of data can cause a dramatic effect in the quality of the received content.

In this paper, as in [19] and our previous work [13], we use Compressed Sensing (CS) [7], [3], [8], [4] for both compression and channel coding of images. 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. Hence, CS can offer an alternative to traditional video encoders by enabling imaging systems that sense and compress data simultaneously. One major advantage to CS encoded data is that the number of unique samples received is the only factor in determining the successful recovery of the image. In other words, *no sample is more important than any other sample* [12]. Because of this, the loss of any single sample can be replaced by another different sample from the same image. The authors of [19] use this property to introduce a method for *oversampling* a signal to increase the chance of recovering a signal that has been subjected to losses. We extend this concept for use with real image signals.

In a real channel, errors within an image transmission will manifest as the inversion of one or more bits within the image signal. These errors can be detected using an Adaptive Parity (AP) scheme [13]. The AP scheme uses a simple parity scheme to determine which samples contain errors. Oversampling and adaptive parity are then used together to find both missing samples and incorrect samples, and compensate for both types using oversampling in a CSEC-AP system. By using this joint system, the transmitting node can both detect and correct bit errors with very little cost to the transmitting node in terms of both complexity and overhead.

In this paper, we propose a system for both determining bit errors in a CS data stream, and consequently compensate for those errors. Specifically:

- **Integrated Adaptive Loss Detection and Compensation.** In our previous work [13], we introduced an AP scheme to find bit errors in a CS data stream. In this work, we combine this concept with oversampling and evaluate an integrated system which can both detect and *compensate for* errored samples.
- **Experimental Evaluation.** We have implemented the AP portion of the protocol on a USRP2 [9] testbed, and are able to show that the results are comparable to those obtained through simulation.

The remainder of this paper is structured as follows. In Section II we give a concise introduction to compressed sensing. In Section III, the joint error detection and oversampling system is presented. In Section IV, we introduce the Compressed Sensed Erasure Coding (CSEC). Section V presents the AP error detection scheme. Finally, the performance results are presented in Section VI, while in Section VII we draw the main conclusions and discuss future work.

II. Compressed Sensing Preliminaries

We consider an image signal represented through a vector $\mathbf{x} \in R^N$, where N is the vector length. We assume that there exists an invertible $N \times N$ transform matrix $\mathbf{\Psi}$ such that

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

where $\mathbf{s}$ is a K-sparse vector, i.e., $||s||_0 = K$ 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. [10]. The signal is measured by taking $M < N$ measurements from linear combinations of the element vectors through a linear measurement operator $\mathbf{\Phi}$. Hence,

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

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 $\mathbf{n} \in \mathcal{N}(\tilde{\mathbf{\Psi}})$ (where $\mathcal{N}(\tilde{\mathbf{\Psi}})$ represents the null space of $\tilde{\mathbf{\Psi}}$), is also a solution to (3). However, it was proven in [3] that if the measurement matrix $\mathbf{\Phi}$ is sufficiently incoherent with respect to the sparsifying matrix $\mathbf{\Psi}$, 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 finding the sparsest solution that satisfies (2), i.e., the sparsest solution that "matches" the measurements in $\mathbf{y}$. However, the problem above is in general NP-hard [2]. For matrices $\tilde{\mathbf{\Psi}}$ 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:

$$P_1 : \; minimize \; ||s||_1$$
$$subject\,to : ||\mathbf{y} - \tilde{\mathbf{\Psi}} \mathbf{s}||_2^2 < \epsilon, \tag{3}$$

where ϵ is a small tolerance.

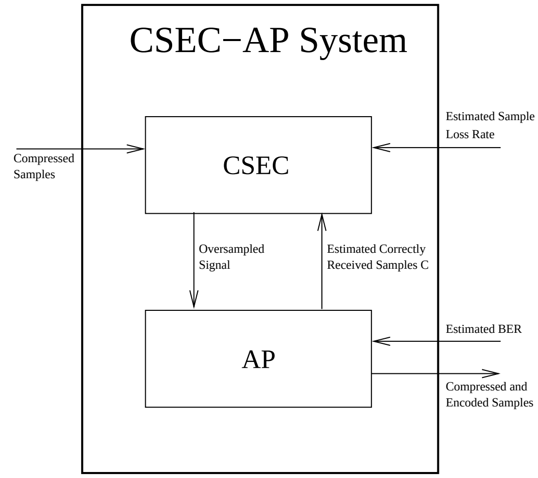

Fig. 1. System Architecture for CSEC-AP System

III. Image Encoding and Recovery using Compressed Sensing with CSEC and AP

We propose a system for image transmission using both CSEC and AP. The architecture for this system if shown in Fig. 1. There are two main goals to this system.

- **Maintain Target Image Quality.** The CSEC portion of the system is charged with maintaining the image quality given a lossy channel. This system takes as input both the number of packets expected to be lost due to collision or transmitter errors and the number of samples expected to be lost due to bit errors that would be detected by the AP system. Oversampling is then used to make up for these errors and allow the receiver to recover the image as if the original number of samples were sent. For example, assume that the transmitter intended to transmit 10,000 samples to the receiver to recover some image. Also assume that 5% of the packets will be lost due to collision or transmission errors, and 3% of the remaining samples will be lost due to bit errors, which results in a total error rate of 7.85%. By oversampling the signal to compensate for the expected loss (as in [19]), the total number of samples K can be found to be 10,852. This tells the transmitter that, based on the loss estimate of 7.85%, if 10,852 samples are transmitted, roughly 10,000 samples will eventually be received correctly at the receiver. The details of the CSEC oversampling rate will be explained in detail in Section IV.
- **Minimize the Number of Transmitted Samples for a Target Desired Quality.** The AP portion of this system uses the *estimated* bit error rate of the channel to determine the optimal number of samples to include for

each parity bit. This system will then use this information to determine the *expected number of correctly received samples*. This is done by analytically determining the optimal number of parity bits needed to maximize the number of correctly received samples at the receiver. The details of the AP calculation will be explained in detail in Section V.

The basis for both of these systems is that the compressed samples which are created using the CS paradigm are all *equally important* and losing a single sample does not affect the receivers ability to be able to recover any other sample. Also, the specific samples chosen for use in the recovery of the image is arbitrary. This means that, if a sample is lost, a different sample can be transmitted in its place with no effect on the quality of the recovered image.

IV. Erasure Channel Coding Using Compressed Sensing (CSEC)

CSEC has the ability to recreate the signal with some degradation even if the errors exceed the threshold for recovery. This is possible by oversampling the signal to compensate for the losses. The total number of samples needed, K, depends on the channel loss probability and is given by

$$K = \frac{m}{(1-p)}, \tag{4}$$

where K is the number of samples needed for a lossless transmission and is a function of the sparsity of the signal and m is the number of correctly received samples samples needed to achieve a desired image quality. Basically, the coding is done such that the number of correctly received samples for a given error probability p is equal to the number of samples in the original signal without errors, i.e. $(1-p)\cdot(K) = m$.

To demonstrate the effectiveness of oversampling, A Monte Carlo simulation of 1000 iterations is performed for a signal of length $256\ \mathrm{byte}$ and of varying sparsity in a channel with a sample loss probability of 0.2. Since the number of samples is the determining factor in the reconstruction of the original signal, there should be no difference between the lossless reconstruction and the oversampled reconstruction. The sampling matrix is incoherent Gaussian. As Fig. 2 shows, as the sparsity increases, the probability of exact recovery of the signal goes down for any channel condition, which corresponds to the results obtained in [19]. Sparsity here is defined as the number of non-zero elements in a signal. This is because as sparsity increases, the information content in the signal increases. If a sufficient number of samples are not generated to compensate for this, all the information conveyed by the signal is not captured and exact reconstruction is not possible. We see that CSEC is able to recover the signal as well as in the case of no loss. This shows that oversampling compensates for the losses in the channel.

Though this shows that oversampling is effective for "ideal" sparse signals, using CS to compress and reconstruct an image could behave differently. This is because an image is not inherently sparse, but is only sparse in the frequency domain

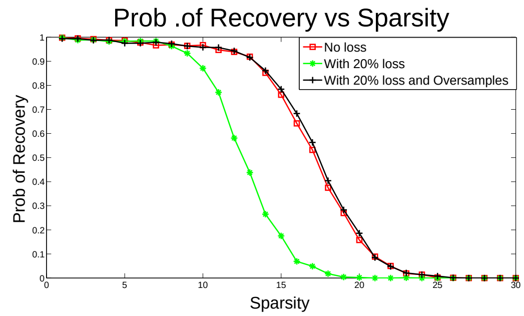

Fig. 2. Probability of exact recovery for recreation of Fig. 8 in [19]

after a wavelet or DCT transform. Any image reconstructed this way will always be different from the original image, and the more samples transmitted, the closer the reconstructed image will be to the original.

To see how the recreation of an image is affected by oversampling, we simulated the recovery of a 32x32 image under three conditions; no loss, 20% sample loss, and CSEC with 20% oversampling. The sampling matrix is assumed to be Gaussian with mean zero and variance $\frac{1}{1024}$. An image size of 32×32 was chosen. The number of measurements in lossless case (m) is taken to be 800. We choose PSNR as the reconstructed image quality indicator, which is defined as

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right), \tag{5}$$

where MAX_I is the maximum possible pixel value for each frame. MSE is the mean squared error, which is defined as

$$MSE = \frac{1}{mn}\sum_{i=1}^{m-1}\sum_{j=1}^{n-1}\|I(i,j) - K(i,j)\|^2. \tag{6}$$

We use the Discrete Cosine Transform (DCT) as the sparsifying transform and CVX to solve the reconstruction problem (3).

In the lossless case, the PSNR is found to be $21.40\ \mathrm{dB}$. With a sample loss rate of 20% and no oversampling, the PSNR drops to $16.78\ \mathrm{dB}$. Finally, with 20% loss and 20% oversampling, the PSNR value is $20.10\ \mathrm{dB}$. Comparing PSNR values of the lossless and oversampled recovery cases, we can see that the images in both cases have similar reconstruction quality. The differences between the errorless case and the oversampling case can be accounted for by variations in the sampling matrix, which was different for each image.

V. Adaptive Parity-based Transmission

For a fixed number of bits per frame, the perceptual quality of images can be improved by dropping errored samples that would contribute to image reconstruction with incorrect information. This is demonstrated in Fig. 3 which shows the image quality both with and without including samples containing errors. Though the plots in Fig. 3 assume that the receiver knows which samples have errors, it does demonstrate that there is a very large possible gain in received image

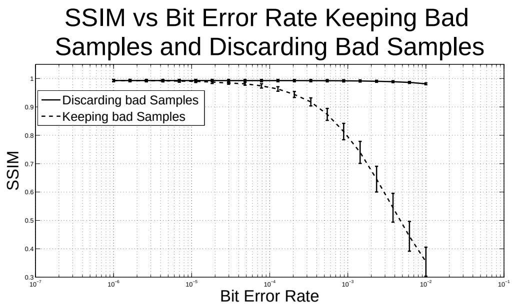

Fig. 3. SSIM for images with and without errored samples

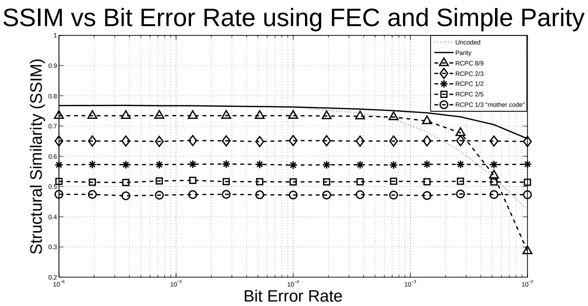

Fig. 4. Adaptive Parity vs RCPC Encoding for Variable Bit Error rates

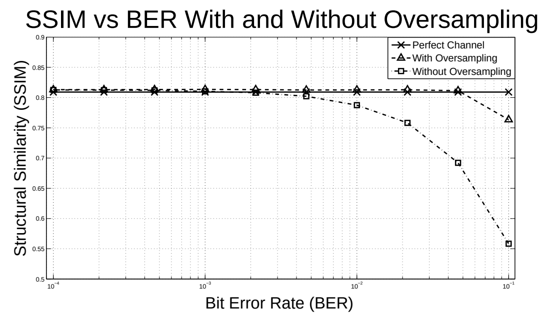

Fig. 5. Performance of CSEC-AP System

quality is those samples containing errors can be found without adding too much overhead.

We studied this for images in [12]. It was shown that in CS, the transmitted 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 only main factor in determining the quality of the received image. Because of this, a sample containing an error can simply be discarded and the impact on the video quality, as shown in Fig. 3, is negligible as long as the amount or errors is small. This can be realized by using even parity on a predefined number of samples, which are all dropped at the receiver or at an intermediate node if the parity check fails. This is particularly beneficial in situations when the BER is still low, but too high to just ignore errors. To determine the amount of samples to be jointly encoded, the amount of correctly received samples is modeled as

$$C = \left(\frac{Q \cdot b}{Q \cdot b + 1}\right)(1 - BER)^{Q \cdot b}, \qquad (7)$$

where C is the estimated amount of correctly received samples, b is the number of jointly encoded samples, and Q is the quantization rate per sample. To determine the optimal value of b for a given BER, (7) can be differentiated, set equal to zero and solved for b, resulting in

$$b = \frac{-1 + \sqrt{1 - \frac{4}{log(1-BER)}}}{2Q}. \qquad (8)$$

The optimal channel encoding rate can then be found from the measured/estimated value for the end-to-end BER and used to encode the samples based on (7). The received video quality using the parity scheme described was compared to different levels of channel protection using rate compatible punctured codes (RCPC). Specifically, we use the $\frac{1}{4}$ mother codes discussed in [11]. Briefly, a $\frac{1}{4}$ convolutional code is punctured to decrease the amount of redundancy needed for the encoding process. These codes are punctured progressively so that every *higher rate* code is a subset of the lower rate codes. For example, any bits that are punctured in the $\frac{4}{15}$ code must also be punctured in the $\frac{1}{3}$ code, the $\frac{4}{9}$ code, and so on down to the highest rate code. Because of this setup, the receiver can decode the entire family of codes with the same decoder. This allows the transmitter to choose the most suitable code for the given data. Clearly, as these codes are punctured to reduce the redundancy, the effectiveness of the codes decreases as far as the ability to correct bit errors. Therefore, we are trading BER for transmission rate.

Figure 4 shows the adaptive parity scheme compared to RCPC codes. For all reasonable bit error rates, the adaptive parity scheme outperforms all levels of RCPC codes. The parity scheme is also much simpler to implement than more powerful forward error correction (FEC) schemes. This is because, even though the FEC schemes show stronger error correction capabilities, the additional overhead does not make up for the video quality increase compared to just dropping the samples which have errors.

VI. Performance Evaluation

We performed two sets of experiments to assess the performance of the proposed error correction architecture. First, a set of images was transmitted using CSEC with AP for different sample loss rates. The image quality is shown for different bit error rates, along with the increase in size necessary to maintain a constant image quality. Secondly, the adaptive parity scheme is tested on a USRP-2 software defined ratio testbed to determine how many errors are correctly detected and the image quality of the detected samples.

A. Simulations

The results shown in Fig. 5 show the reconstruction of images encoded using the CSEC-AP system, only the AP

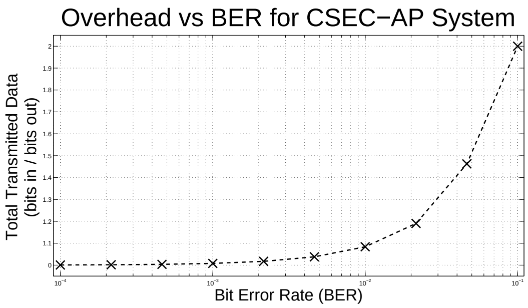

Fig. 6. Overhead of CSEC-AP System

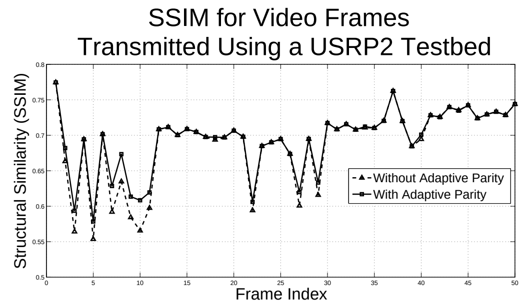

Fig. 7. Results of Testbed Implementation of CSEC-AP system

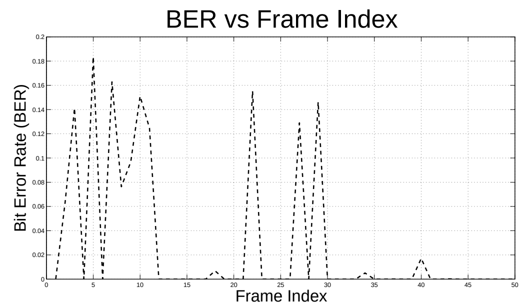

Fig. 8. Measured BER in Testbed Experiments

system (detecting and simply removing bad samples) and with an ideal lossless situation. Clearly, the proposed system results in image quality very near to the ideal no-loss case at reasonable BER rates. In all cases, it is assumed that a sample is 8 bits and that even a single bit error within a sample results in that entire sample being discarded. The cost in terms of overhead for this error correction scheme is shown in Fig. 6, which shows the additional transmitted information as a function of the bit error rate. We can see that even for the worst case of 1 error for every 10 bit (resulting in a sample error rate of 0.5695), the CSEC-AP scheme only requires the transmitter to send twice the required bits for a channel without errors. As the error rate drops down to more reasonable values, the overhead decreases very quickly.

B. Testbed Experiments

The adaptive parity portion of the scheme was also tested using a USRP2 software defined radio platform. The performance of the algorithm was evaluated on a testbed that comprised of USRP2s. A two hop network was setup to evaluate the performance of the algorithm.

The medium access control (MAC) layer protocol selected was 802.11b and differential quadrature phase shift keying (DQPSK) was used as the modulation scheme to achieve a physical layer data rate of $2\ \mathrm{Mbit/s}$. The maximum size of each packet was $2100\ \mathrm{bytes}$. The packets were transmitted in burst mode with each burst consisting of at most six packets. At the transmitter, a parity bit was appended for a certain number of samples that was determined from the current bit error rate of the channel and on the encoding of the frame using (8). A $100\ \mathrm{byte}$ header for bit error estimation preceded the data in the packet. Transmissions were made on a selected frequency in the $2.4\ \mathrm{GHz}$ ISM band. A CSMA/CA scheme with random backoff time was implemented at the MAC layer to alleviate the effects of packet collisions. The relay node has a queue structure that simply forwards the received packets to the destination node. The receiver after receiving the packets decodes the data and determines which samples were corrupted during transmission based on the parity bit. If there is a parity bit inversion, all the samples that were included in that parity bit calculation are dropped. Also, the receiver estimates the bit error rate of the channel through the $100\ \mathrm{byte}$ header. The transmitter, after obtaining the estimate of the bit error rate, calculates the new number of samples/bit and the cycle continues. We assume the initial bit error rate of the channel to be zero. So, the samples transmitted during the first burst will not have any parity bits appended.

This system was used to transmit and decode 50 frames of a security video. The results of this simulation are shown in Fig. 7, while the measured BER is shown in Fig. 8. Even with very high bit error rates, the algorithm was still able to recover the images nearly as well as predicted by the simulations. Whenever there were sample errors, the results using AP were better than those using no error correction at all.

VII. Conclusion and Future Work

In this paper, we have presented a system which uses compressed sensing to compress an image and protect that image from channel errors and packet losses. We have expanded on the work done in [19] and [13] to present a complete system which deals with the detection of bit errors and provides a system for compensating for these bit errors in such a way as to maintain image quality at the receiver. We also presented a testbed setup using USRP2 software defined radios. We implemented a portion of the system on the testbed and demonstrated that the performance is very close to the simulation results.

Future plans for this work include expanding the use of CS encoded images to video encoding. Also, we are using the properties of CS encoded images in other networking layers such as the transport and MAC layers in order to create a system which will be able to transmit video from very simple low-cost image sensors. This system will be tested using the USRP2 testbed.

REFERENCES

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

[2] A. Bruckstein, D. Donoho, and M. Elad. From Sparse Solutions of Systems of Equations to Sparse Modeling of Signals and Images. *SIAM Review*, 51(1):34–81, February 2009.

[3] E.J. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, February 2006.

[4] E.J. Candes and T. Tao. Near-optimal Signal Recovery from Random Projections and Universal Encoding Strategies? *IEEE Transactions on Information Theory*, 52(12):5406–5425, December 2006.

[5] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, July 2006.

[6] D. Schmidt, M. Berning, and N. Wehn. Error Correction in Single-Hop Wireless Sensor Networks-A Case Study. In *Proc. of Design, Automation and Test in Europe (DATE)*, Nice, France, April 2009.

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

[8] E.J. Candes and J. Romberg and T. Tao. Stable Signal Recovery from Incomplete and Inaccurate Measurements. *Communications on Pure and Applied Mathematics*, 59(8):1207–1223, August 2006.

[9] Ettus Research LLC. http://www.ettus.com/.

[10] A. Graps. An Introduction to Wavelets. *IEEE Computational Science and Engineering*, 2:50–61, 1995.

[11] J. Hagenauer. Rate-compatible punctured convolutional codes (RCPC codes) and their applications. *IEEE Transactions on Communications*, 36(4):389–400, Apr 1988.

[12] T. Melodia and S. Pudlewski. A Case for Compressive Video Streaming in Wireless Multimedia Sensor Networks. *IEEE COMSOC MMTC E-Letter*, 4(9), October 2009.

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

[14] I. S. Reed and G. Solomon. Polynomial Codes Over Certain Finite Fields. *Journal of the Society for Industrial and Applied Mathematics*, 8(2):300–304, 1960.

[15] A. Viterbi. Convolutional Codes and Their Performance in Communication Systems. *IEEE Transactions on Communication Technology*, 19(5):751–772, October 1971.

[16] Viterbi, A. Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, Apr 1967.

[17] Viterbi, A.J. and Wolf, J.K. and Zehavi, E. and Padovani, R. A Pragmatic Approach to Trellis-Coded Modulation. *IEEE Communications Magazine*, 27(7):11–19, Jul 1989.

[18] Stephen B. Wicker. *Error Control Systems for Digital Communications and Storage*. Prentice-Hall, Inc., New Jersey 1995.

[19] Z. Charbiwala and S. Chakraborty and S. Zahedi and Y. Kim and M. B. Srivastava and T. He and C. Bisdikian. Compressive Oversampling for Robust Data Transmission in Sensor Networks. In *Proc. of IEEE Conf. on Computer Communications (INFOCOM)*, San Diego, CA, March 2010.