Compressive Sampling for Energy Efficient and Loss Resilient Camera Sensor Networks

Ashish Wani and Nazanin Rahnavard
Department of Electrical and Computer Engineering
Oklahoma State University, Stillwater, OK 74078
Emails: {ashish.wani,nazanin.rahnavard}@okstate.edu

***Abstract*—Data loss is inevitable in multi-hop wireless sensor networks. Multiple packet erasures during transmission can necessitate the use of error-control mechanisms for loss recovery. Energy consumption is also very critical in embedded sensor networks. These problems are even more severe in wireless camera sensor networks (WCSNs), owing to the large data size. Compressive Sampling (CS) turns out to be an effective solution on both the issues. The compression obtained through the linear projections allows transmission of lesser bits than the original. The inherent randomness in CS makes the system tolerant to losses without requiring transmission of redundant parity bits. Both these characteristics help us on saving up on energy. However, using conventional CS on embedded WCSNs has some implementation related challenges. WCSNs mainly find applications in surveillance systems. This requires the snapshots to be large enough to encompass a wide field of view; requiring image sizes at least QVGA or more. The processor memory and the recovery time of L1 optimization, needed for CS recovery, are non-linear with respect to the data size; hence large images hinder the applicability of CS in practical cases.**

In this paper, we address the issues which may affect the practical usability of CS and provide a CS framework suitable for WCSNs. In order to enable the processing of such large images we propose a block-wise sampling approach, which helps to reduce both the memory overhead and the recovery time. For sampling matrices we use binary sparse random matrices instead of dense matrices, so as to reduce the encoding and decoding times and the computational overheads. Moreover, to further reduce the time factor we employ very sparse matrices (row weight equal to one) and show that they still provide good quality images. We have tested our propositions on WCSNs that include Imote2 sensor nodes equipped with IMB400 multimedia boards, on which we have analyzed the loss resilience of our proposed framework and have also provided an estimate of the energy saved.

I. Introduction

WCSNs are embedded systems capable of capturing and transmitting snapshot or streaming multimedia over a multi-hop network. These features make them suitable for surveillance and monitoring; applications which demand longevity of operation while maintaining a certain degree of quality of service. The different challenges in design and implementation of multimedia sensor networks have been summarized in [1].

Power consumption has been a fundamental concern in traditional wireless sensor networks (WSNs) and is even more critical in WCSNs with the voluminous multimedia data adding to the strain on the transmission costs. This makes *efficient* transmission schemes a necessity for WCSNs to operate for long durations. Data loss has been a notorious problem in the wireless domain and it gets aggravated in sensor networks because of the harsh environments and a limited power source. Excessive losses can even make complete transmissions useless. Forward error correction (FEC) and automatic repeat request (ARQ) schemes are available to combat erasures, but they could be complex and impose large overheads.

The emerging field of compressive sensing (aka compressive sampling) [2], [3], turns out to be an effective solution to both these problems and also suitable for implementation in the sensor networks domain. The theory of compressive sampling enables us to generate samples of a sparse signal through *linear random projections* such that the compressed signal is much smaller than the original. This signal can be recovered using the fact that the L1 minimization of a CS signal can provide the sparsest solution. This helps to mitigate the challenge of reducing the transmission costs. Because of the inherent randomness in the sampling stage, random erasures make little difference to the overall signal statistics making the CS data resilient to losses.

On the other hand, using CS for WCSNs has some practical difficulties. WCSNs are being developed with prospective applications in surveillance and monitoring. This requires the cameras to capture images over a wide field of view, which also requires large images to be processed. It is known that the recovery algorithms of CS data such as the Basis Pursuit have a complexity of $O(N^3)$ [4], where N is the length of the original signal. This makes the compression and recovery of large images computationally intensive with long recovery times. These factors hinder the use of conventional CS in practical WCSN applications and hence new approaches need to be introduced.

In this paper we attempt to address issues which hinder the applicability of CS on an actual WCSN system and provide solutions to these problems through a CS framework implemented on an Imote2 Wireless Sensor node and a IMB400 Multimedia board. We test our framework and suggest optimum CS parameters that can be used. Also, we analyze its loss resilience and provide a probable estimate of the energy saving.

II. Related Work

As mentioned before the inherent randomness and signal compression can solve both the limited power and data loss problems simultaneously, making CS an ideal choice for WSNs. Performance of CS for erasure coding has been discussed in [5].In this work the authors propose a Compressive oversampling approach to compensate for the expected erasure to maintain a target signal quality. This work has been extended in [6], where the oversampling is performed for expected loss due to bit-errors too, along with channel erasure, leading to some improvement in the reconstruction quality. The premise of this Oversampling approach is that the target signal quality is known. The problem here is that in a practical WCSN application, determining a target quality is not possible, because it is a no-reference system. Also, a standard compression ratio at the transmitter cannot be assumed because through erasure analysis we found that the number of samples required for a given reconstruction quality varies with the type of image captured. Thus, the compression at the camera will depend on the image being captured and hence will vary as per application.

Conventional CS schemes suggest a dense random projection matrix for signal sampling. However, it was shown later that binary and sparse random matrices have a good performance as well and are very convenient for implementation. [7] and [8] have a very good analysis on binary sparse random matrices and discuss the special properties of binary sparse projection matrices, especially the impact of the sparsity of the matrix (low column weight in their case), in the reconstruction as well as the recovery time. The findings in these studies can be related to our work for theoretical analysis. [9] proposes a *block compressed sensing* approach for improving the recovery time and memory storage. The approach in [9] involves using a sampling matrix similar to an FIR filter to generate CS measurements from the traversed portion of the image. For reconstruction, a minimum mean squared error estimation is used to obtain an initial linear estimation and then use some non-linear techniques for refinement. Using this approach in sensor networks is difficult because of the filter type implementation to generate samples, which increases the computation time and cost. Moreover, the performance of the linear estimation in presence of erasures is not known.

In this paper we provide solutions to aforementioned problems and present a CS framework that is suitable for implementation on an actual embedded sensor node.

III. Compressive Sampling for Wireless Camera Sensor Networks

A. Compressive Sampling Overview

The theory of compressive sampling defines sampling as $y = \Phi x$, where x is a signal of length n that is sparse over some proper basis, y is the measurement vector of length $m << n$ and Φ is the sampling (projection) matrix, which generates the random linear projections. At the recovery phase, x can be recovered by solving an l1 minimization [3].

Images are not sparse in the spatial domain, but they are sparse in some other transform domains such as the wavelet or DCT. Hence, image x is made sparse by s= $\Psi^T x$, where s is the sparse representation of x and Ψ is an orthonormal basis. If these transformations have to be performed on a sensor node then it would be detrimental in terms of power consumption and operational latency, because of the added computations. Here CS holds an advantage as the data need not be sparse while sampling; measurements can be taken in the original dense domain (i.e., $y = \Phi x$). It is only at the receiver that the transform basis needs to be known. The signal can be reconstructed by finding s (and accordingly $x = \Psi s$)from the system of linear equations $\underline{y} = \Phi\Psi\underline{s}$. This is an *underdetermined* system with infinitely many solutions. However, the sparsity of $\underline{s}$ allows us to have a successful reconstruction with a high probability. It was shown in [2] that $\underline{s}$ can be estimated via solving the following ℓ_1 optimization problem

$$\hat{s} = \arg\min \parallel s \parallel_{l_1} \; s.t. \qquad y = \Phi\Psi s. \tag{1}$$

B. Selecting CS parameters

1) Measurement Matrix Φ: The main condition that a measurement matrix has to satisfy to get the best samples from a signal is the Restricted Isometry Property (RIP) [3]. There are a number of matrices which satisfy this property, some which are dense matrices such as the Gaussian random matrices, constructed by selecting i.i.d random variables from a Gaussian distribution. Same is the case with the Bernoulli matrix. These matrices have a very good performance but the measurements will be expensive to generate on a resource-constraint embedded processor. Hence, another class of matrices called Sparse Binary matrices are used. Sparse binary matrices have a performance comparable to dense matrices as shown by [7]. Their compression process comprises of random selection and addition of a small subset of signal components, hence they are easy and fast to compute. This makes sparse matrices better than the dense matrices for embedded applications. The total number of signal components added to generate one measurement is equal to the weight of the corresponding row in the sampling matrix. For our implementation we use a constant row binary matrix.

2) Orthonormal Basis Ψ: As mentioned earlier, the signal can be compressed in its dense domain, just that the domain of sparse transformation should be known at the receiver. But from (1) we can see that the orthonormal (transform) basis needs to be available in its matrix form during the recovery. This constraint arises because we are sampling in the dense domain. The problem with this additional constraint is that, higher dimensional wavelet transforms are not available in a matrix form and hence cannot be used. The basis that can be represented in a matrix form are the 1D Haar, Discrete Cosine Transform (DCT) and Discrete Fourier Transform (DFT).

There is another property of CS that a pair of measurement matrix and orthonormal matrix have to satisfy for a good quality reconstruction; the mutual coherence. The coherence between the measurement matrix (Φ) and orthonormal matrix (Ψ) is defined as,

$$\mu(\Phi, \Psi) = \sqrt{n}_{1 \le i \le m, 1 \le j \le n} max \; |< \Phi_i, \Psi_j >|, \tag{2}$$

where Φ_i's are rows of Φ and Ψ_j's are columns of Ψ. Thus, the coherence measures the correlation between the rows of Φ and the columns of Ψ. In CS low coherence value pairs of Φ and Ψ are required [10]. Therefore, even though the selection of the orthonormal basis depends on its ability to sparsify the original signal, it also depends on the sampling matrix used; for optimum results. The binary sparse matrix is highly coherent with the 1D Haar matrix; hence even though an image has a better sparse representation in the Haar wavelet, it cannot be used. Between DCT and DFT, the DCT performs better than DFT because of the better energy compaction. Hence, we employed the DCT matrix as the orthonormal basis.

Mutual Coherence also helps in setting up another parameter, the row weight of the sparse matrix. As seen earlier the row weight will determine how many pixels are combined together to generate one sample. Intuitively, it seems like a higher row weight will ensure more original signal elements being combined and hence a better result; however, this is not entirely correct. The results with a higher row weight are better, but as the row weight increases, so does the coherence between the DCT and Sparse sampling matrix as shown in Figure 1. Because of these two opposing effects, the improvement in quality with row weight is not significant

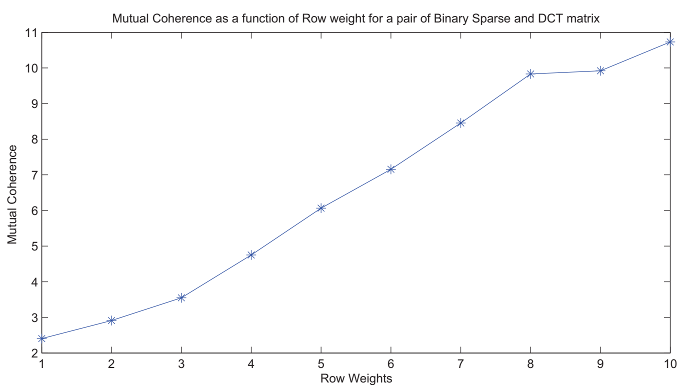


Fig. 1. Mutual coherence vs row weight of the measurement matrix

C. Block-wise Sampling

From (1) it is clear that Ψ is required to be known at the base station for the exact recovery of the original signal. The Ψ being used here is a DCT matrix. Suppose an image has a width of w and height h. The DCT matrix Ψ will be of size $w * h \times w * h$. In a double precision format, this matrix requires $w^2 * h^2 * 8$ bits of memory storage. This is a very huge requirement, considering that the smallest practically used size is the QVGA(320x240), which may require about 6 Gigabytes of Random Access Memory(RAM) Storage. Another problem that arises from large image sizes is the recovery time of ℓ_1 optimization. The recovery time holds a non-linear relationship with the vector size it operates on. For a size of QVGA, this time may go in the order of 2-3 hours. This makes CS incompetent for practical implementation. Also, these problems will become exponentially worse as image sizes increase.

Using a 2D DCT is an option, but it puts a constraint of using only square images. Hence, as a solution to these problems, we propose a *block-wise* sampling scheme, where samples are taken from the image vector in a block-wise manner. Measurements taken from each block are independent of the other block. Since the vector length reduces, both problems of memory and recovery time are solved. But the quality of reconstruction depends on the length of the block; larger the size, better the image quality [3]. Hence, there is a trade-off between the image quality and the recovery time, for which both the parameters need to be optimized.

IV. Implementation

A. On the camera sensor node (transmitter)

We implement our proposed schemes on an Imote2 sensor node along with the IMB400 multimedia board, which bears the camera. The image type is gray scale, hence a uint8 data type is sufficient to store a single pixel. Each packet comprises of a header field and a 64 byte (64 pixel) payload. For compressive sampling, two operations have to be performed to generate a measurement; first random selection of pixels and second addition of those selected pixels. For random selection, a pseudo-random generator is implemented using a Multiplicative Congruential Generator. The default generators were not used in order to use the same generators at the transmitter and the receiver. A multiplicative congruential generator is simple to implement, while generating a good pseudo random sequence, hence it is suitable for implementation on an embedded platform. A seed is a number fed to the pseudo random number generator at the start of each generation cycle, which ensures the repeatability of the random sequence. We use the *packet numbers* as the seeds to ensure uniqueness. After random selection, these pixels are added together, up to the row weight; to generate one measurement. These measurements are then packetized and this data is sent to the next hop node.

B. At the base-station

At the base-station we need to recover the original signal from the compressed signal. For that we need to know the information about the random combinations of the *recieved* measurements. In order to obtain the correct combinations, two conditions need to be satisfied. The first is to use of the same pseudo random generator used at the transmitter and the second is to provide the correct seed to the generator. We comply with the first condition by implementing the same multiplicative congruential generator on MATLAB. For the second condition, we use the packet number as the seed. Use of the packet number as the seed ensures uniqueness in the seed values and also no extra seed values need to be conveyed through the payload. From these seeds we construct a *sampling reconstruction matrix* (Φ') for the received data. This matrix is different from the one at the transmitter since, because of the losses, the transmitted data is not same as the received one. The matrix is constructed from the fact that each measurement corresponds to one row and each randomly generated number corresponds to a column. Each row is filled until it reaches its row weight. Thus, we build the matrix (Φ') from the received data. The dimensions of (Φ') are m'xn, where m' is the number of *received* measurements. This received signal is recoverable because, due to the inherent randomness, in spite of the losses the overall statistics of the matrix do not change. It is this property that contributes to the loss resilience of CS. Now with the sampling matrix and the orthonormal matrix available, the original signal can be recovered from the measurements using l1 optimization as given in equation (1).

C. Block-wise sampling

The only information available about the image array is its starting address. The task is to take CS samples from each block independently. Segmenting the image vector into an actual block-wise structure would be cumbersome and the process will have a high latency factor. A faster and simpler way of block-wise sampling is to select from a range within the block size instead of the image size and the addressing done with respect to the block's starting address. This ensures that only one block is sampled at a time. Once samples are taken from the block, the block start address is incremented by an amount equal to the block size, ensuring that the next set of selections are made from the next block. Thus, block-wise sampling is performed by virtue of simple additive computations,which take the least processing power.

V. Experiments and Analysis

As mentioned earlier, the sampling part has been implemented on an Imote2 sensor node.The image is compressively sampled for transmission over the wireless channel. The radio chip used, CC2420 has a CRC (Cyclic Redundancy Check) check field. If the CRC check fails, the packet is marked and the TinyOS module discards the packet. Thus, the channel can be modeled as a Packet Erasure Channel.

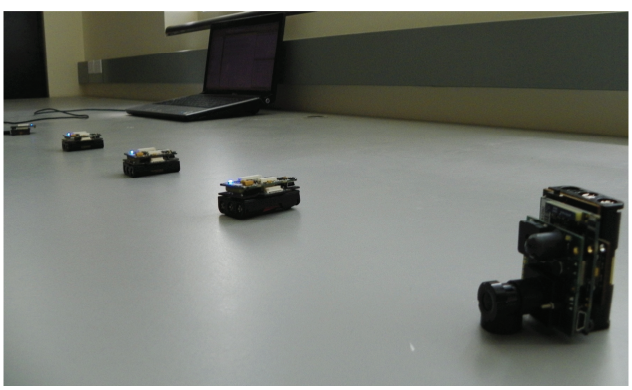


Fig. 2. Experimental Setup

The base-station receives the data and it is read into MATLAB using its serial reader interface. The recovery of CS data has been done using the Basis Pursuit implemented using MATLAB 7.1(R2010a) on an Intel Xeon 3GHz processor. The SPGl1 package is used for *l1*-norm minimization.

For analyzing the quality of reconstruction of the images, we use the Structural Similarity Index (SSIM) as the metric. This is a widely used reference based image quality assessment technique [11]. To obtain a reference image, we first take an uncompressed no loss snapshot of the object under consideration. All the future reconstructed images are compared against this reference image.

A. Block size vs Quality and Recovery Time

In this experiment we show the effect of the block size and row weight on the quality of reconstruction and recovery time. A longer block gives better quality while a smaller one takes less memory and has a lower recovery time.

The experiment has been done on a 128x128 Lenna image, since this is the largest size of single image that could be computed without block-sampling, as more memory was required to store the large Ortho-normal matrix.

The number of samples taken are 50 percent of the original image. Through these results it can be seen that the blocks size has a huge impact on the recovery time, due to the non-linear relationship. Similarly, the row weight is also a major factor in the recovery time. The important thing here is that the difference in quality between the best case result (block size 1, row weight 10) and the worst case (block size 16, row weight 1) is tolerable, but with significant gain in time. Thus, depending on application and requirement these parameters can be selected.

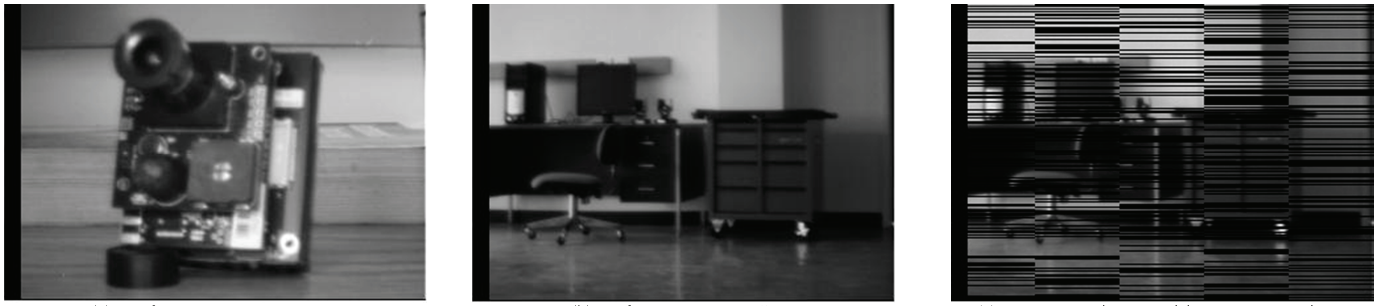
TABLE I
PERFORMANCE FOR VARIABLE BLOCK SIZE AND ROW WEIGHT

No.of blocks	SSIM Rwt 10	SSIM Rwt 1	Time(s) Rwt 10	Time(s) Rwt 1
1	0.9221	0.9203	1600	625
2	0.9215	0.9168	1514	236
4	0.9234	0.9149	974	181
8	0.9164	0.9100	228	68
16	0.9168	0.9004	78	30

B. Reconstruction performance with variable row weight on real image

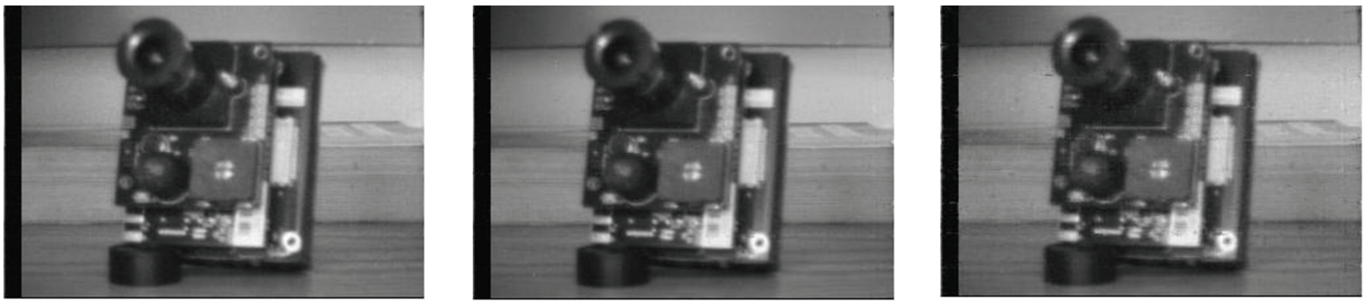
The results of increasing row weight can also be seen for the images obtained from the sensor cameras.The reference image is shown in fig. (3a). Since the object is close to the camera, the image is dense, with a number of details being captured. The quality of the reconstructed image can be observed from the number of details preserved.

The original size of these images is 320x240(QVGA), hence the length of the image vector is 76800 pixels. These images have been taken at 70 percent (53760) measurements of the total image size, with a block size of 7680 pixels per block. As seen in the reconstruction performance, figure (4), there is almost no perceptive difference between the images. Their SSIM values are 0.9512, 0.9396 and 0.9239 and the time taken in minutes is 145.9, 88.2 and 27.6 for row weights 5,2 and 1 respectively. From these results it is clear that with row weight, the quality does not improve much but there is significant gain in time. This makes a very sparse binary matrix (row weight=1) a very useful proposition.



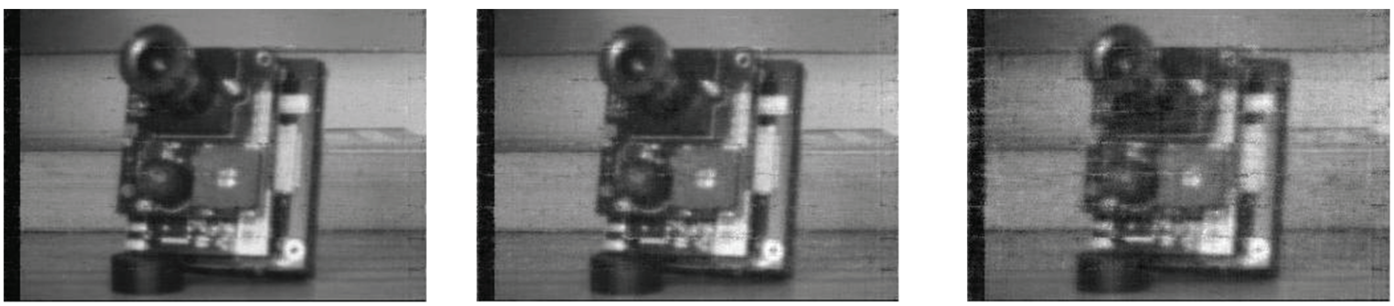
(a) Reference Image - Dense (b) Reference Image- Sparse (c) Erasure on image without compression

Fig. 3. Reference images



(a) Row weight 5 (b) Row weight 2 (c) Row weight 1

Fig. 4. Reconstruction Performance with variable row weight



(a) Low, 10 percent (b) Medium, 30 percent (c) High, 60 percent

Fig. 5. Erasure performance for dense image

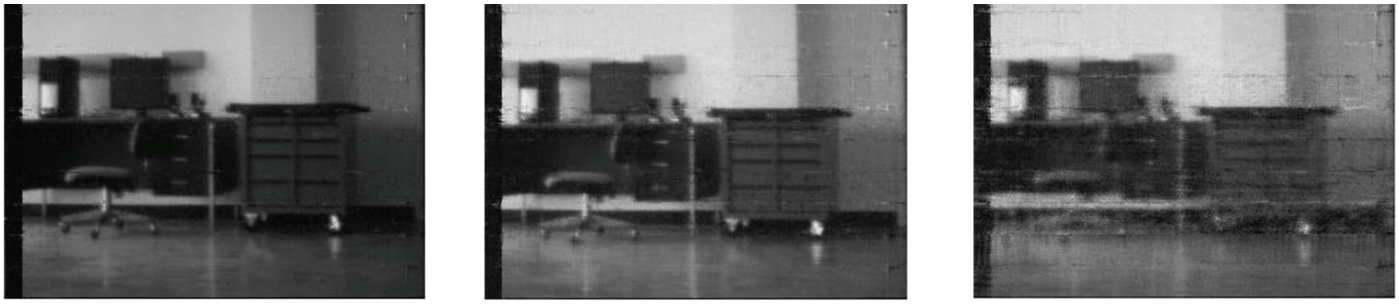
C. System performance in case of erasures

In this experiment we test the loss resilience of the sensor network system. For this we use one camera sensor taking snapshot data. It passes this data through a series of intermediate nodes to the base-station. The packets are routed using a simple address based scheme. Even though intermediate nodes act just as transceivers, packet erasure is observed because of the CRC check drop and ambient noise. The variation in erasure was obtained by changing the number of hops and the distance between the nodes, keeping the transmission power constant.

The experiment has been performed on two types of images, one image is a close-up snapshot of an object, we call this a spatially dense image fig. (3a). This image contains a lot observable features, which can be referred for subjective analysis. We can observe the system performance as per the reconstruction of these features. Another image is one capturing a wider area, hence the image is spatially sparse fig. (3b). The image reconstruction results can be seen in figures (5) and (6). The low degree is 10 percent erasure, moderate 30 and high being 60 percent erasure. These images have been taken at initial compression of 70 percent, so the received data will be much less, for example,the effective percentage data received for 60 percent erasure, will be around 21 percent of the image size. Considering that such small amount of data is received the reconstruction quality is good, especially as compared to the performance of the uncompressed image fig. (3c). From both the image types we can observe that even for moderate erasures the system performance is very good. This is better analyzed through a plot of the SSIM's of the reconstructed images as a function of the erasure shown in fig. (7). From the plot we can see that the sparse image has a smoother degradation, even though its original quality is a bit less. This shows that the reconstruction performance also depends on the type of image under consideration.

D. Energy Saved

We call CS an energy efficient mechanism. The approximate savings for one image transmission are: if we transmit at 3dBm



(a) Low, 10 percent (b) Medium, 30 percent (c) High, 60 percent

Fig. 6. Erasure performance for sparse image

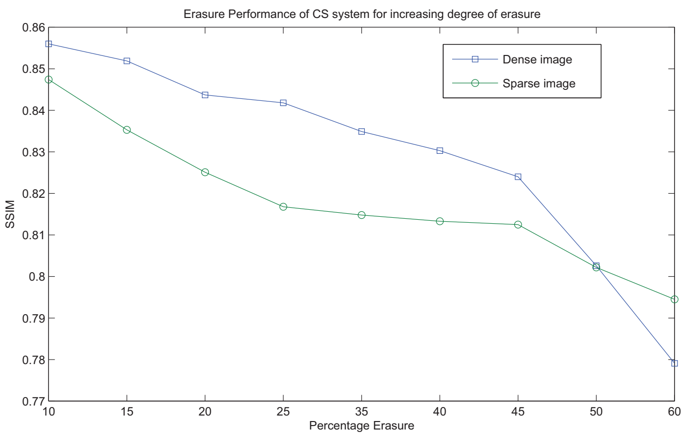


Fig. 7. Performance of CS for increasing erasure

power level at which the Energy/bit for transmission is approximately 105nJ/b [12] then at 70 percent compression, we transmit only 53760 bytes. Therefore, bits reduced with respect to original (76800-53760)*8= 184,320 bits. This corresponds to approximately 19 mJ per node per transmission.

The Imote2 has a DSP co-processor along with main processor, making it difficult to calculate the consumption per instruction cycle. But the instruction overhead imposed by the multiplicative congruential generator used for CS includes two extra modulus operations, two additions and 1 multiplication. The processors used are known for very low power operation hence these additional instructions should not consume much and a sizable energy conservation should be expected over a large network.

VI. Conclusion

In this paper we propose a framework for implementing Compressive Sampling on a WCSN platform,such that it will bring CS closer to real time application requirements. As part of the framework we suggest the use of very sparse binary random matrices (row weight 1). They are ideal for hardware implementation, reduce the number of bits transmitted and also help to reduce the recovery time. We also suggest a block-wise sampling approach to reduce the recovery time and the storage memory requirements at the receiver. We support our propositions through experimental results which show that our proposed system does not degrade the quality of reconstruction by much, but the recovery time reduces significantly. Thus our framework overcomes some important factors that hinder the practical implementation of CS on WCSNs.

VII. Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. ECCS-1056065.

References

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

[2] D. Donoho, "Compressed sensing," *Information Theory, IEEE Transactions on*, vol. 52, pp. 1289 –1306, april 2006.

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

[4] R. Baraniuk, "Compressive sensing [lecture notes]," *Signal Processing Magazine, IEEE*, vol. 24, pp. 118 –121, july 2007.

[5] Z. Charbiwala, S. Chakraborty, S. Zahedi, Y. Kim, M. B. Srivastava, T. He, and C. Bisdikian, "Compressive oversampling for robust data transmission in sensor networks," in *Proceedings of the 29th conference on Information communications*, INFOCOM'10, (Piscataway, NJ, USA), pp. 1190–1198, IEEE Press, 2010.

[6] S. Pudlewski, A. Prasanna, and T. Melodia, "Resilient image sensor networks in lossy channels using compressed sensing," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on*, pp. 62 –67, 29 2010-april 2 2010.

[7] R. Berinde, A. Gilbert, P. Indyk, H. Karloff, and M. Strauss, "Combining geometry and combinatorics: A unified approach to sparse signal recovery," in *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*, pp. 798 –805, sept. 2008.

[8] A. Gilbert and P. Indyk, "Sparse recovery using sparse matrices," *Proceedings of the IEEE*, vol. 98, pp. 937 –947, june 2010.

[9] L. Gan, "Block compressed sensing of natural images," in *Digital Signal Processing, 2007 15th International Conference on*, pp. 403 –406, july 2007.

[10] E. Candes and J. Romberg, "Sparsity and incoherence in compressive sampling," *INVERSE PROBLEMS*, vol. 23, pp. 969–985, JUN 2007.

[11] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *Image Processing, IEEE Transactions on*, vol. 13, pp. 600 –612, april 2004.

[12] N. Chohan, "Hardware assisted compression in wireless sensor networks abstract."