On-Mote Compressive Sampling in Wireless Seismic Sensor Networks

Marc J. Rubin
Computer Science Ph.D. Candidate
Department of Electrical Engineering and Computer Science
Colorado School of Mines
mrubin@mines.edu

Problem and Motivation

Implementing a wireless sensor network capable of near real-time, continuous (e.g., 500 Hz sampling rate), geophysical sensing requires a better approach than a simplistic sense, store, send methodology. Transmitting all sensor data is very power intensive, as the radio consumes much more power than other wireless mote components (e.g., CPU, ADC) [1]. To address the need for a lightweight data reduction algorithm, we investigate the viability of on-mote compressive sampling (CS), both in simulation on seismic data and in real hardware.

Background and Related Work

The relatively new subject area of compressive sampling (or sensing) was first proposed in the mid 2000s. Compressive sampling seeks to replace the current notion of "sample *then* compress" with "compress *while* sampling". Briefly, we summarize how CS works [3, 4, 5, 6, 7, 8, 9, 10].

Many real-world signals are sparse with regard to some basis. Sparsity implies that a signal can be represented by only a few non-zero coefficients. For example, a sinusoid wave (x) is not sparse in the time domain, but sparse in the frequency domain (Ψ) with few non-zero coefficients (α), i.e., $x = \Psi\alpha$. CS makes use of sparsity to estimate the original signal ($\widehat{x}$) from very few randomly selected measurements (y). In particular, it uses convex optimization to solve the underdetermined linear system:

$$\widehat{x} = argmin_{x'}||x'||_{\ell_1} \quad \text{subject to} \quad y = \Phi x'.$$

The measurement matrix, Φ of size $M \times N$, is responsible for "compressing" x, the original signal of length N, to a more manageable measurement vector y, of length M, using matrix multiplication (with M $\ll$ N). There are several types of measurement matrices, including random Gaussian (e.g., Φ = randn(M,N) in Matlab) and random binary, which has M rows and N columns, with exactly one 1 per row (with zeros elsewhere). In a random binary measurement matrix, the ones are randomly distributed throughout the N columns in increasing order, such that the index of the one in row $m-1$ is less than the index of the one in row m. In our simulations, we use a random binary measurement matrix because it can be implemented very efficiently in hardware.

CS is a non-adaptive compression algorithm, meaning that the rate of compression is *not* directly dependent on signal redundancies. The non-adaptive nature of CS is particularly appealing for wireless sensor networks because the rate of compression and thus, radio usage, can be specified by the user and power savings can be guaranteed throughout the lifetime of wireless data acquisition. The same cannot be said for adaptive compression algorithms, where the rate of compression is directly dependent on signal redundancies; thus, the rate of compression can be roughly estimated but not guaranteed.

Approach and Uniqueness

In our experiments, we first simulated six combinations of sparsity domains and compressive sampling recovery algorithms on five-minute chunks of real-world seismic data (sampled at 500 Hz) containing 33 slab avalanches events (e.g., Figure 1a). The data was collected in the mountains outside of Davos, Switzerland during the 2010-2011 winter season [11]. The six combinations tested three different signal recovery algorithms: 1) ℓ_1-norm minimization (L1), 2) orthogonal matching pursuit (OMP) [12], and 3) reweighted ℓ_1-norm minimization (RWL1) [13], each with assumed sparsity in the frequency (Fourier) or time-frequency (Gabor) domains. Furthermore, for each recovery algorithm and sparsity domain, we tested a number of compressive sampling percentages, from 10% to 90% of the original (full) signal. Note that the compressive sampling percentage is inversely proportional to the rate of compression; for example, 30% compressive sampling, where $M = 0.3N$, is equivalent to 70% signal compression. We defined the compression rate (or ratio) to be:

$$CompressionRatio = 100 \times (1 - \frac{CompressedSize}{OriginalSize}).$$

Additionally, we implemented a novel CS algorithm on an Arduino Fio wireless mote, recorded an artificial sinusoid signal produced by a signal generator, and measured the mote's power consumption during algorithm execution. Our algorithm, called Randomized Timing Vector (RTV) [14], essentially computes $y = \Phi x$ without performing a costly on-mote matrix multiplication. In other words, instead of collecting the full signal x of length N and calculating $y = \Phi x$, we simply sample y, of length M, directly. This is accomplished by transforming a random binary measurement matrix Φ_b (size $M \times N$) to a vector, V_t of length N with M randomly placed ones (with zeros elsewhere). For example:

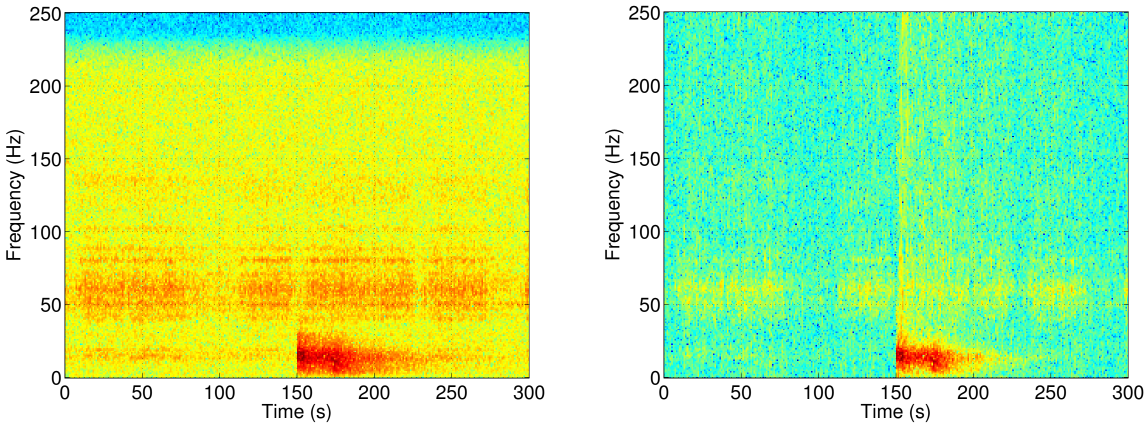
$$\Phi_b = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \Longrightarrow V_t = \{0, 1, 0, 1, 1\}.$$

This timing vector is stored in the mote's memory, and indicates which M of every N samples to sample, store, and transmit. Specifically, every two milliseconds (500 Hz sampling rate) the mote checks the timing vector indexed at the current sample number (modulo N), and senses, stores, and transmits the data if and only if a one is present. Lastly, the transmitted vector y is used to recover the original audio signal offline using ℓ_1-norm minimization assuming sparsity in the frequency domain. Our preliminary results provide a novel implementation of on-mote CS to record a signal, reduce radio transmissions, and thus, save power.

Lastly, we compared CS to five other lossy and lossless compression algorithms designed specifically for resource limited wireless hardware. In particular, we compared CS to two lossless compression algorithms, namely Lempel-Ziv-Welch for sensor nodes (S-LZW) [15] and run-length encoding (RLE) [16]. In terms of lossy algorithms, we evaluated K-run length encoding (K-RLE) [16], lightweight temporal compression (LTC) [17], wavelet quantization thresholding RLE (WQTR) [18], and CS [19, 20, 14].

Results and Contributions

Results from our simulations show that CS is a viable data reduction technique for wireless seismic sensor networks. Figures 1a and 1b show frequency spectrograms of an original and one of the recovered slab avalanches. In both figures, the slab avalanche occurred at time 150 seconds and had significant low frequency energy. Qualitatively, the slab avalanche event is clearly present in the recovered signal, which was acquired using 30% simulated compressive sampling (70% compression).



(a) 100% sampling

(b) 30% compressive sampling, RWL1 recovery from Gabor domain

Figure 1: (a) A spectrogram of the original signal of slab #13 with 100% full sampling. (b) A spectrogram of the recovered signal from 30% compressive sampling (70% compression ratio) via RWL1 recovery assuming sparsity in the Gabor time-frequency domain.

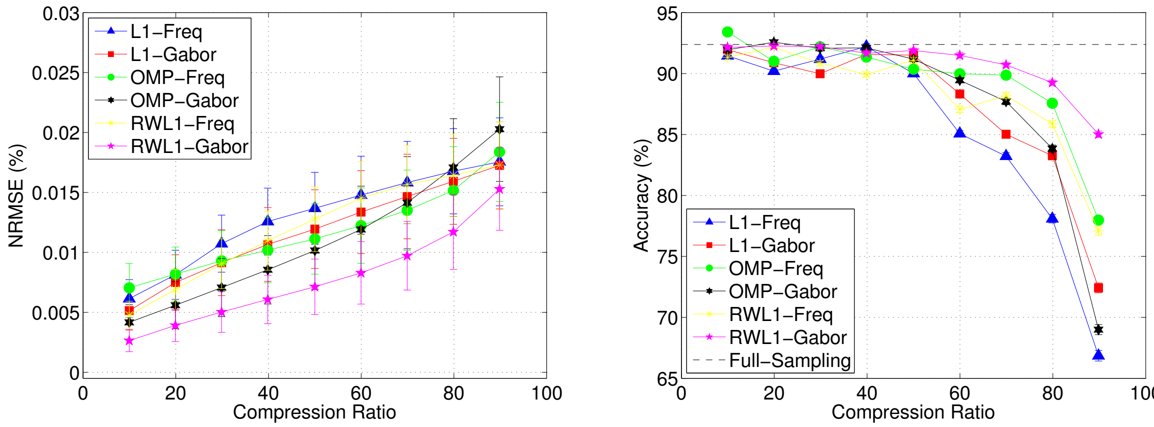
For a more quantitate analysis, we first calculated the normalized root mean square error between the original and recovered signals. NRMSE is calculated as:

$$NRMSE = \frac{\sqrt{mean((x-\widehat{x})^2)}}{max(x)-min(x)} \times 100,$$

where $\widehat{x}$ is the recovered signal and x is the original signal containing a slab avalanche. Our results (Figure 2a) show that RWL1-Gabor, i.e., reweighted ℓ_1-norm minimization recovery algorithm assuming sparsity in the time-frequency (Gabor) domain, had the least recovery error. After recovering the seismic data containing 33 slab avalanches, we tested our machine learning workflow to automatically detect avalanche events [21]. The results are excellent; for example, with signals recovered from 30% of the data (70% compression), we were able to obtain 90.7% classification accuracy (compared to 92.4% accuracy with full sampling or 0% compression).

Next, we analyzed the power consumption of our lightweight and novel on-mote RTV compressive sampling algorithm against full sampling. In other words, we measured the current consumption of an Arduino Fio wireless mote running our RTV algorithm (with various compressive sampling rates) and full sampling while it acquired an artificial sinusoid signal. The results in Figure 3 clearly shows the significant power savings that CS offers. Specifically, CS can offer up to a 5-fold increase in battery longevity compared to full sampling.

Lastly, our comparison of six on-mote compressive algorithms revealed that, in terms of recovery error, CS performs comparably (Figure 4a). Specifically, CS was the second best performing lossy compression algorithm and fell within the 95% confidence intervals of K-RLE, the best performing lossy algorithm. These results are striking, considering that CS is the only non-adaptive compression algorithm evaluated, meaning that compression rates are not directly dependent on signal redundancies. Instead, the rate of compression can be specified beforehand by the user and



(a) (b)

Figure 2: (a) The mean NRMSE with 95% confidence intervals for the six combinations of CS tested on real-world seismic data containing avalanches. (b) The mean classification accuracy with 95% confidence intervals for the six combinations of CS evaluated on our real-world seismic data..

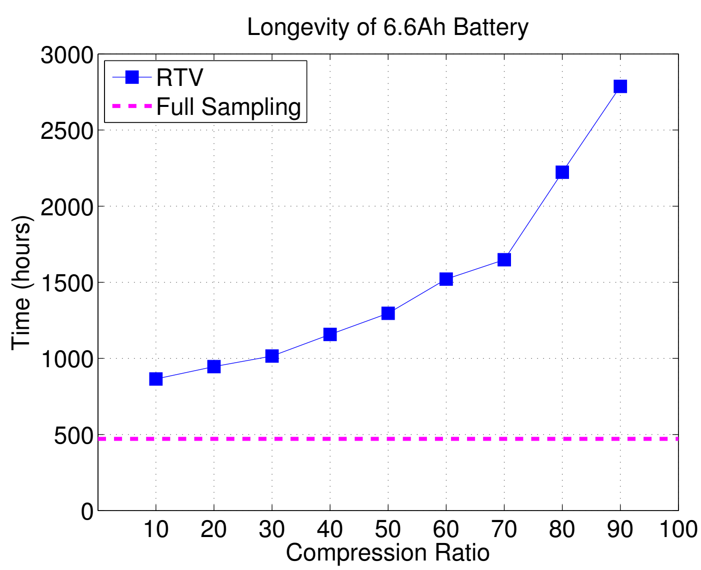
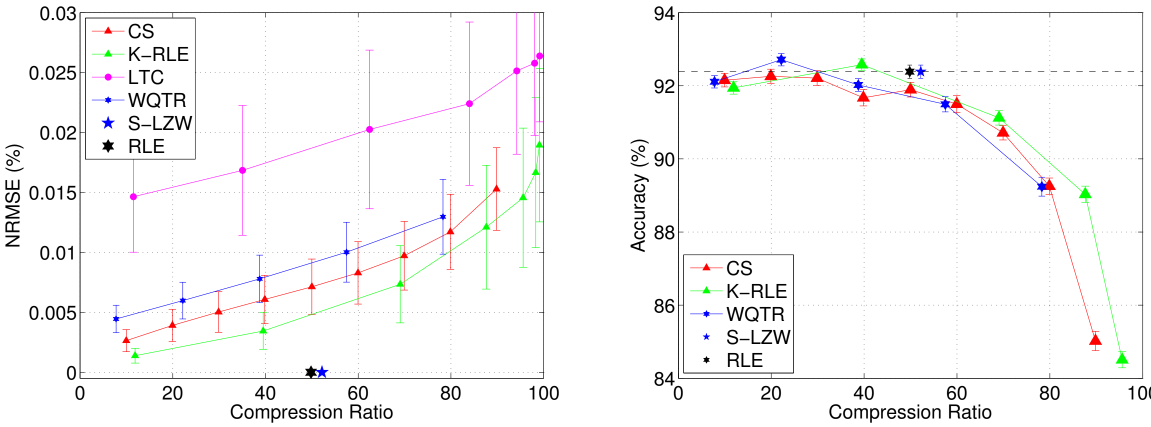


Figure 3: Power consumption (in terms of battery longevity) of the three on-mote CS algorithms running on an Arduino Fio wireless mote.

guaranteed throughout the lifetime of wireless data acquisition. Similarly, avalanche classification results show that CS performed reasonably well, with accuracies 0.1 to seven percent below classification accuracies that use the entire original signal (Figure 4b). As noted previously, the accuracy of detecting slab avalanche events from 30% of the data (70% compression) was 90.7%. For comparison, with S-LZW lossless compression, we reached 92.4% classification accuracy with 52% compression. In other words, we had 18% more compression for CS (compared to S-LZW) but suffered only a 2% decrease in classification accuracy.

The main contribution of this work shows that compressive sampling is a viable data reduction technique for wireless seismic data acquisition. First, we showed that using the RWL1 recovery



(a)

(b)

Figure 4: (a) The mean NRMSE with 95% confidence intervals for the six compression algorithms tested. (b) Mean classification accuracies with 95% confidence intervals for five of the six compression algorithms evaluated. LTC was omitted due to significantly poor recovery error rates.

algorithm assuming sparsity in the time-frequency (Gabor) domain had the least recovery error and highest classification accuracies. Second, we showed that our on-mote compressive sampling algorithm, called RTV, can significantly decrease wireless node power consumption and increase battery longevity. Lastly, we showed that CS performed comparably to other state of the art wireless compression algorithms; in terms of NRMSE, CS was the second best performing lossy compression algorithm, falling with the 95% confidence intervals of K-RLE, the best performing lossy algorithm.

The results presented are quite promising, considering that CS is a non-adaptive compression algorithm where the rate of compression can be specified by the user before sensor node deployment. In other words, the rate of compression, radio usage, and thus, power savings, can be guaranteed throughout the lifetime of wireless data acquisition.

References

[1] N. Kimura and S. Latifi, "A survey of data compression in wireless sensor networks," *IEEE International Conference on Information Technology: Coding and Computing (ITCC)*, 2005.

[2] T. Srisooksai, K. Keamarungsi, P. Lamsrichan, and K. Araki, "Practical data compression in wireless sensor networks: A survey," *Journal of Network and Computer Applications*, vol. 35, pp. 37–59, 2012.

[3] E. Candes, "Compressive sampling," *International Congress of Mathematicians*, vol. 3, 2006.

[4] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, 2006.

[5] D. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[6] E. Candes and M. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, March 2008.

[7] R. Baraniuk, "Compressive sensing," *IEEE Signal Processing Magazine*, vol. 24, pp. 118–121, July 2007.

[8] D. Mackenzie, "Compressed sensing makes every pixel count," *What's Happening in the Mathematical Science, American Mathematical Society*, vol. 7, pp. 114–127, 2009.

[9] M. Davenport, M. Duarte, Y. Eldar, and G. Kutyniok, *Introduction to Compressed Sensing: Chapter in Compressed Sensing: Theory and Applications*. Cambridge University Press, 2012.

[10] M. Fornasier and H. Rauhut, *Compressive Sensing: Chapter in Part 2 of the Handbook of Mathematical Methods in Imaging*. Springer, 2011.

[11] A. Herwijnen and J. Schweizer, "Monitoring avalanche activity using a seismic sensor," *Cold Regions Science and Technology*, vol. 69, no. 2-3, 2011.

[12] J. Tropp and A. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 53, no. 12, 2007.

[13] E. Candes, M. Wakin, and S. Boyd, "Enhancing sparsity by reweighted l1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, 2008.

[14] M. Rubin and T. Camp, "On-mote compressive sampling to reduce power consumption for wireless sensors," *IEEE International Conference on Sensing, Communication, and Networking (SECON)*, 2013.

[15] C. Sadler and M. Martonosi, "Data compression algorithms for energy-constrained devices in delay tolerant networks," *ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2006.

[16] E. Capo-Chichi, H. Guyennet, and J. Friedt, "K-RLE: A new data compression algorithm for wireless sensor network," *IEEE International Conference on Sensor Technologies and Applications (SENSORCOMM)*, 2009.

[17] T. Schoellhammer, E. Osterweil, B. Greenstein, M. Wimbrow, and D. Estrin, "Lightweight temporal compression of microclimate datasets," *IEEE International Conference on Local Computer Networks (LCN)*, 2004.

[18] N. Xu, S. Rangwala, K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, "A wireless sensor network for structural monitoring," *ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2004.

[19] Z. Charbiwala, Y. Kim, S. Zahedi, J. Friedman, and M. B. Srivastava, "Energy efficient sampling for event detection in wireless sensor networks," *ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED)*, 2009.

[20] H. Mamaghanian, N. Khaled, D. Atienza, and P. Vandergheynst, "Compressed sensing for real-time energy-efficient ECG compression on wireless body sensor nodes," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 9, 2011.

[21] M. Rubin, M. Wakin, and T. Camp, "On-mote compressive sampling in wireless geophysical sensor networks," tech. rep., Colorado School of Mines, 2012.