

Wavelet Compression with Set Partitioning for Low Bandwidth Telemetry from AUVs

Chris Murphy
Woods Hole Oceanographic Institution
Woods Hole, MA, USA
cmurphy@whoi.edu

Hanumant Singh
Woods Hole Oceanographic Institution
Woods Hole, MA, USA
hsingh@whoi.edu

ABSTRACT

AUVs typically communicate with scientists on the surface over an unreliable acoustic channel, resulting in very low data throughput. While there are several examples of scientific data, even imagery, being successfully transmitted over high rate acoustic links, channel coding methods with high rates of error-correction are often employed that limit data throughput to tens or a few hundred bits per second. Little research exists into appropriate methods for image and data compression for acoustic links at these very low rates.

We recently have experienced great success using compression techniques based upon the Set Partitioning in Hierarchical Trees (SPIHT) embedded coding method, and feel they are particularly suited to underwater data. In particular, SPIHT provides a fully embedded coding method; truncating the encoded bitstream at any point produces the optimal encoding for that data length. This allows fine-resolution imagery to build on previously transmitted low-resolution thumbnails. For time-series data, we have developed a method for quantizing data to emphasize more important sections, such as the most recently collected data.

In this paper we describe how these methods can be applied to compress scalar environmental data and imagery for communication over acoustic links. We also present initial results of sea trials performed near Rota in the Commonwealth of Northern Marianas Islands, during which images were captured, compressed and transmitted in-situ.

Categories and Subject Descriptors

I.4.2 [Image Processing]: Compression(Coding)

General Terms

Algorithms, Design, Experimentation, Measurement

Keywords

Autonomous Vehicles, Data compaction and compression

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WUWNet'10, Sept. 30 - Oct. 1, 2010, Woods Hole, Massachusetts, USA
Copyright 2010 ACM 978-1-4503-0402-3 ...\$10.00.

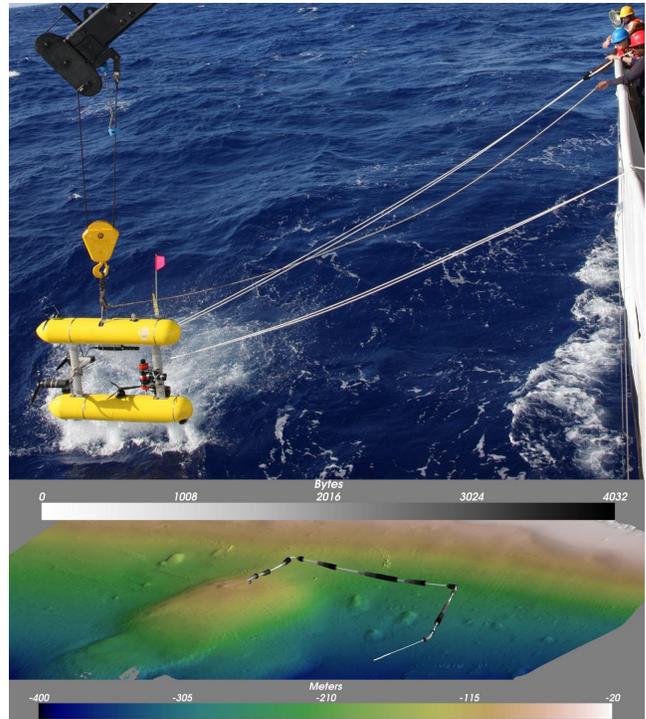


Figure 1: At top, SeaBED AUV prior to launch near Rota, CNMI, about 1500 miles south of Tokyo; AUV mission track overlaid on bathymetry at bottom.

1. INTRODUCTION

Free of a physical tether to the surface, Autonomous Underwater Vehicles (AUVs) are able to reach and explore areas that would be hazardous, or impossible, with more traditional tethered or human occupied underwater vehicles. AUVs have explored the bottom of Antarctic glaciers and Arctic ice sheets, and points dotting the oceans between. In each of these missions, the AUV returned with data to be analyzed by surface operators, before being sent back down for additional missions. There remains no substitute for the high-level decision making skills of a human operator when it comes to mission planning.

Unfortunately, the freedom to operate without a physical tether comes at a cost. The ocean imposes severe limitations on wirelessly communicating data to the surface including low available bandwidth and long propagation delays [23,

1]. AUV and surface ship noise combine with environmental conditions to cause a host of problems including frequent packet loss. These challenges are made worse by operating over large distances and by environmental conditions such as seafloor makeup and water depth.

Effective data rates for acoustic modems used to communicate underwater are routinely as low as tens of bits per second. Connections may be unpredictably intermittent, with long periods of no communication. Time-multiplexing of the channel for Long Baseline (LBL) navigation, or round-robin communication of multiple vehicles, lowers effective bit-rates for each vehicle even further.

Over the course of a dive, a single AUV can easily collect one million samples of scalar environmental data, ranging from temperature data to salinity, and from measured methane concentrations to vehicle depth. The same vehicle may easily capture tens of thousands of photos, and sonar imagery. This sensor data is typically inaccessible until after the vehicle has been recovered, since low transmission rates have largely limited vehicle telemetry to vehicle state and health information. Temperature and reduction potential (Eh) data acquired during two SeaBED AUV dives, shown in Figure 2, are used to illustrate compression within this paper.

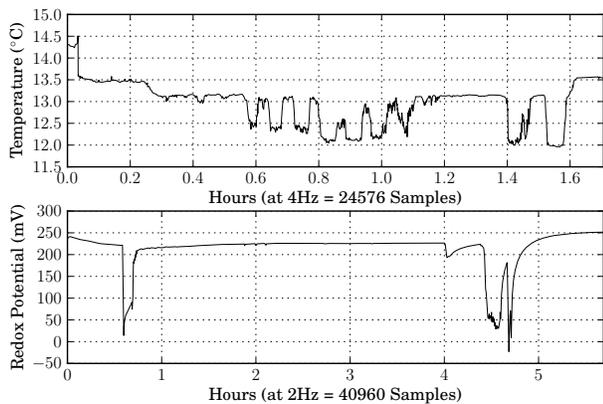


Figure 2: Sample scalar environmental data. Temperature data was collected over an archaeological site near Santa Barbara, California using a SeaBED AUV. The reduction potential data was collected as part of the Arctic Gakkel Vent Expedition [12], and provided by Dr. Koichi Nakamura.

As autonomous robotics moves toward the study of dynamic processes with multiple vehicles, there is an increasing need to compress the large volumes of data collected by AUVs for transmission to human operators, and to other vehicles. Vehicle recoveries from the open ocean are challenging and risky for both vehicles and operators. Additionally, vehicles may take hours to ascend from missions in the deep sea. Methods for retasking autonomous vehicles during a deployment with new high-level mission goals have begun to be developed and tested on multiple vehicles [11, 28].

While the underwater community has previously investigated transmission of imagery and video data over high-throughput acoustic links, there has been a relative lack of algorithmic transfer from the data compression community for low-throughput environments. In this paper, we present

a systems level approach to data compression, well suited for use with AUVs. We evaluate this approach against scalar data from two different sensors and dives, shown in Figure 2. Finally, we show recently acquired results of applying these same methods, on-line, to images captured by a SeaBED-class[22] AUV operating near Rota in the Commonwealth of the Northern Mariana Islands (CNMI).

2. PREVIOUS WORK

Compression techniques are divided into two categories; lossless compression methods that allow faithful reconstruction of the original data, and ‘lossy’ methods which allow reconstruction of an approximation. There are numerous general-purpose lossless algorithms, such as LZ77[29] and Lempel-Ziv-Welch (LZW)[27]. While generic lossless algorithms perform well with plain text and other widely used document formats, they do not perform particularly well on numerical scientific data[27].

As a result, techniques specifically for compressing floating-point scientific data have been developed. However, even recent lossless algorithms yield on the order of 2:1 compression for high entropy time-series data[2]. Compressing the example data using each of the methods mentioned above actually resulted in significantly better compression ratios from the general algorithms in one case than from the floating-point specific algorithms, as shown in Table 1.

Method	Redox Potential		Pot. Temp.	
	Size (B)	Ratio	Size (B)	Ratio
Raw data	115200	—	57208	—
gzip	29770	387.0%	38173	149.9%
bzip2	24300	474.1%	40318	141.9%
FPZip	77460	148.7%	35475	161.3%
FPCompress	88345	130.4%	41820	136.8%

Table 1: Comparison of compression ratios for several lossless compression methods. FPCompress[2] and FPZip[13] were evaluated using source code from the authors’ websites. Two hours of reduction potential data sampled at 2Hz, including the segment shown in Figure 2, and a two-hour section of potential temperature data sampled at 1Hz (not shown) were used for the comparison.

One widely used standard for underwater vehicle communications is the Compact Control Language (CCL)[24], which defines a number of simple methods for encoding individual samples of depth, latitude, bathymetry, altitude, salinity, and other data. CCL relies only upon quantization to provide compression and makes no use of the inherent correlation between successive samples from most instruments.

In 1996, Eastwood et al. proposed predictive coding methods that could be used in concert with these methods to improve performance[5]. Schneider and Schmidt have incorporated predictive coding into their recent work[20], sending up a mean value followed by smaller, quantized, delta values. While this provides some compression, transform codes allow higher efficiency for a bit more computational effort.

Transform compression methods typically follow a standard pattern. First, a source coder such as the Discrete Cosine Transform (DCT) or Discrete Wavelet Transform

(DWT) exploits the inherent correlation within most data, and concentrates the energy of the signal into a sparse set of coefficients. Next, these coefficients are quantized and entropy encoded [16]. Wavelet compression is described by Donoho et al. as being especially appropriate for functions that are “piecewise-smooth away from discontinuities” [4]. While not all sensors emit signals of this form, this is an apt description for many oceanographic sensors.

There has been extensive experimentation with the transmission of still and video imagery over relatively high bandwidth (~1-10kbps) acoustic tethers[25, 15]. In addition, there has been some previous study on the application of wavelet compression techniques to underwater images, video, and acoustic imagery[9, 10, 8]. In particular, Eastwood et al. evaluated the performance of an early Wavelet-based compressor, EPIC, in 1996[5]. Craig Sayers, and others at the University of Pennsylvania, developed techniques for selecting specific frames and ‘regions of interest’ from a video sequence that best describe an ROV manipulator and environment state, and transmitted these regions to surface operators over a 10 kbps acoustic tether as JPEG images[19].

3. APPROACH

Said and Pearlman developed and first introduced SPIHT in the late 1990’s[17]. By coupling their coding method with the DWT, SPIHT can be used as a data compression technique for both time-series scalar telemetry and imagery. SPIHT has several characteristics that make it particularly suited to compression of underwater data. In particular, the algorithm is straightforward and can be performed efficiently on general purpose processors, and provides high compression ratios for a variety of real-world data. If high framerate processing is necessary, high-speed hardware implementations of the image compressor have been developed [7].

More uniquely, SPIHT provides a fully embedded coding method; truncating the encoded bitstream at any point produces the optimal encoding for that data length. This feature is not shared by all, or even most, compression methods; half of a zip file does not allow easy restoration of half the contents, and half of a JPEG image is not immediately viewable at reduced quality. This makes it well suited to the underwater environment where computation ability is limited, communication is packetized, and transmission rates can vary from packet to packet, as it allows compression to be performed independent of the target transmission rate. Messages sent to nearby AUVs for multiple vehicle collaboration could be sent at a higher rate, and those destined for a surface ship or transmission over longer distances can be sent at a more conservative rate without any need for recompression of the data. Low fidelity color image thumbnails, transmitted at rates as low as a few hundred bits per image, can later be used as a basis for more refined versions. If the entire bitstream is sent, the compression process is entirely reversible and results in the original data with no loss of precision.

SPIHT can be used effectively on scalar data, imagery, or even 3D volumetric data. For simplicity, we will discuss the one dimensional approach first, and then extend to imagery. Our approach to data compression consists of three discrete steps. First, data is encoded using the DWT into the wavelet domain. Next, these (typically floating-point) coefficients are requantized as signed fixed point numbers.

Finally, this fixed-point representation is encoded using the SPIHT algorithm, which results in a sequence of bits. This result is truncated to the desired length and transmitted. Once received, the truncated bitstream is simply decoded into a signed fixed point approximation to the wavelet coefficients. The Inverse DWT is then performed on these coefficients, resulting in an approximation to the original data.

3.1 Discrete Wavelet Transform (DWT)

Effective source encoders concentrate most of the energy of the original signal into a smaller number of coefficients. These coefficients will no longer be correlated across different input sequences, as they could then be compressed further [18]. The DWT, a linear transform, is widely used as a source encoder for image and biomedical data. For a well-written introduction to wavelets, DeVore and Lucier provide an excellent reference [3].

The DWT is calculated by applying a low-pass filter to the input signal, generating one set of coefficients, and then applying a high-pass filter to the input signal to generate a second set of coefficients. Both sets of coefficients are downsampled by two, resulting in the same number of coefficients as the original input signal had samples. Calculating the DWT of a signal thus results in two distinct sets of coefficients; a decimated version of the signal known as the ‘approximation coefficients’, and a set of ‘detail coefficients’ which contain the higher-frequency information lost during decimation. Figure 3 shows the full wavelet decomposition of a short signal of 32 samples.

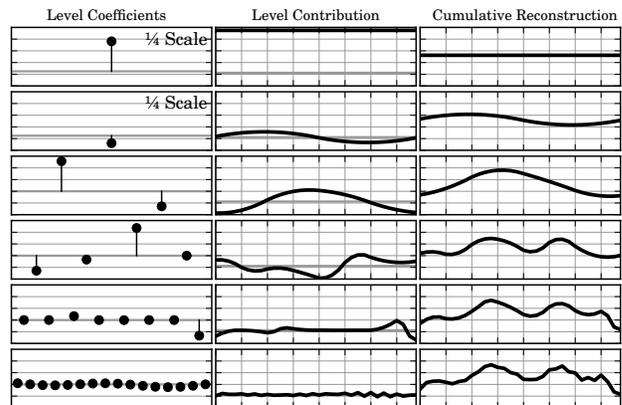


Figure 3: Wavelet coefficient magnitude is shown by the stem plots at left. The middle column indicates the sum of the inverse transformed wavelets at that level of detail. By cumulatively summing the levels (right column), increasingly detailed approximations to the original signal are produced until the original signal is recovered at the bottom right.

The DWT is typically (as in Figure 3) applied recursively to the approximation coefficients, generating several levels of detail coefficients; each level of detail coefficients then represents the detail lost by decimation at that iteration of the transform. Each detail coefficient in the resulting set is localized in time as well as being associated with a ‘scale’, or level of detail. The detail coefficients will generally be low

in magnitude, except near areas of change for a given scale. This sparsity facilitates efficiently compressing the data.

3.2 SPIHT Coding

While the SPIHT algorithm is straightforward to implement, it has a large number of details addressed by the authors in their initial paper and in a later book chapter [14]. As we believe our implementation to be a faithful implementation of their description, we do not seek to convey all details of the algorithmic implementation. This section instead provides an intuitive understanding of how SPIHT works, and what information the resulting bitstream contains. Details of our field trial are discussed in Section 4.2.

SPIHT, and its progenitor the embedded zerotree wavelet (EZW) [21] algorithm, treat the wavelet decomposition as a tree of coefficients, rooted at the lowest level detail coefficients. Many real signals that have large magnitude coefficients at high levels also have higher magnitude coefficients at lower levels. SPIHT exploits this cross-level correlation through a clever sorting algorithm. As the SPIHT authors write in their tutorial[14, p95] set partition coding

...is a procedure that recursively splits groups of [coefficients] guided by a sequence of threshold tests, producing groups of elements whose magnitudes are between two known thresholds.

A SPIHT-encoded bitstream consists of a sequence of refinement bits and sorting bits, interlaced in a data-dependent order. Specifically, there are five things that a single bit in a SPIHT bitstream could represent:

- Sorting Bits
 - Whether a coefficient is greater in magnitude than the current threshold, or ‘significant’
 - Whether any descendant is significant
 - Whether any grand-descendant is significant
- Refinement Bits
 - The sign of a coefficient
 - A single bit of a coefficient’s magnitude

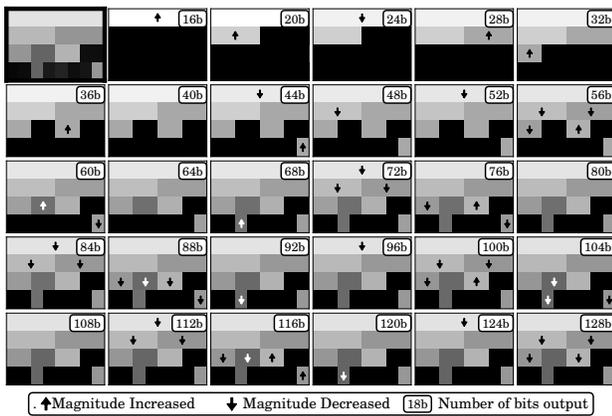


Figure 4: A wavelet decomposition at upper left, followed by the reconstruction from increasingly length SPIHT bitstreams. As the number of bits grows, the reconstruction is closer to the original coefficients. Coefficient signs have not been depicted.

Refinement bits provide a continually improving estimate for the magnitude of a wavelet coefficient. Sorting bits provide an efficient way to identify high magnitude, and therefore important, wavelet coefficients. Figure 4 shows the progressive reconstruction of a set of DWT coefficients using an increasing number of (indicated) bits.

3.3 Image Compression

SPIHT was originally designed for photo compression, and can be used on high dimensional datasets as well. Two-dimensional data like imagery is simply transformed with the 2D form of the DWT, and then SPIHT coded following a similar process as the 1D version. To encode color images, we first transformed each image to the Y’UV color space. Each color plane was then coded separately to a fixed number of bytes, with the U and V color planes receiving a much smaller allowance than the luminance plane.

4. RESULTS

Encoding the scalar data with SPIHT resulted in a significant improvement in data fidelity across a wide range of transmission rates, when compared to simple subsampling. The received signal is both qualitatively, and quantitatively (RMS error), more similar to the original data than interpolated data points, as shown in Figure 5. As a side-effect,

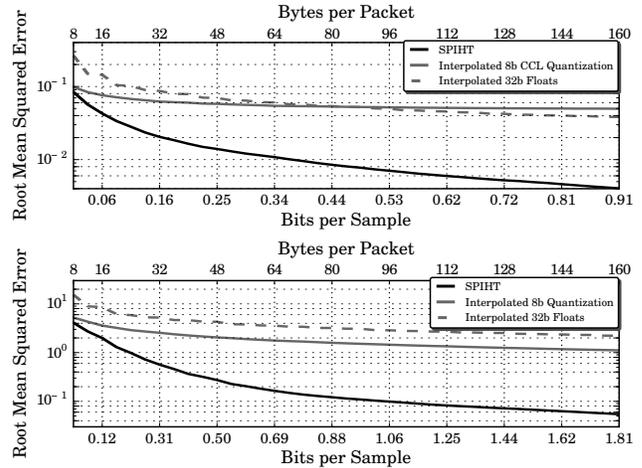


Figure 5: Comparison of SPIHT encoding with subsampling methods for Temperature data (top) and reduction potential data (bottom).

the reconstructed signal has been de-noised; discarding low-magnitude coefficients is an effective form of noise reduction [26].

In real-world implementations for in-situ encoding, there are two key parameters that can be adjusted during encoding of scalar data. The number of bytes to send up in each packet (throughput), and the number of data samples to encode in each time (frequency of transmission). Assuming a fixed frequency of 1024 samples (approximately 5-10 minutes between transmissions in this case) yields the encoding results shown in Figure 6. Decreasing the transmission frequency can greatly increase the compression ratio, as shown in Figure 7.

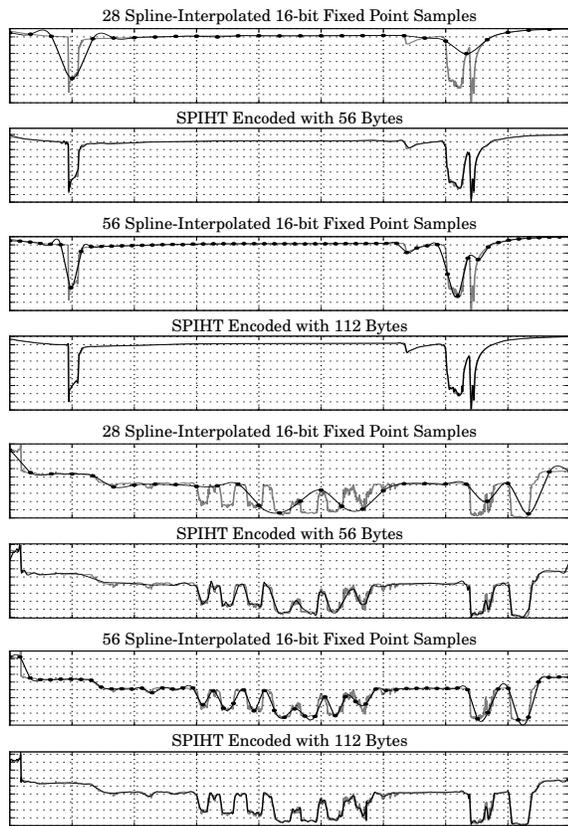


Figure 6: Comparison of signal reconstruction at two different sizes, for both test signals.

4.1 Time-Varying Quantization

In [17] wavelet coefficients are requantized into a standard sign-magnitude representation before SPIHT coding. What level and method of quantization is appropriate for a given signal depends upon the dynamic range, maximum and minimum values, and acceptable level of error for a time-series; the quantization is typically constant for all wavelet coefficients.

Occasionally, it may be of value to provide higher fidelity to certain sections of scalar data. Accenting recent data would allow decisions to be made about nearby features of interest before they are left far behind. Additionally, if the time windows for transmitted data overlapped somewhat from transmission to transmission, missing data due to packet loss could be filled in with future, lower fidelity, data.

One approach that holds some promise is to vary the quantization method for the wavelet coefficients using some cost function. This strategy has been employed to generate Figure 8; wavelet magnitudes were artificially prescaled prior to encoding them with SPIHT. As SPIHT prioritizes higher magnitude coefficients, this leads to greater detail being conveyed in those areas. Upon decoding, reverse scaling is applied before the inverse wavelet transform.

Figure 8 was generated using the logarithmically increasing sequence of quantization coefficients shown in Equation 1, where n is the number of coefficients. In the Python pro-

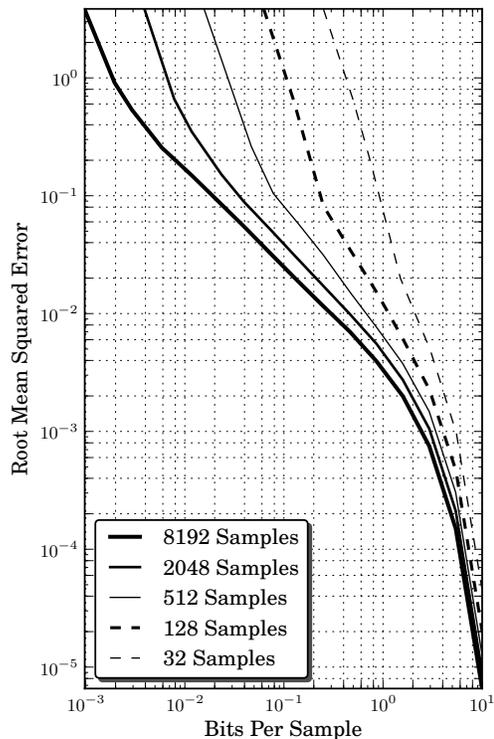


Figure 7: Encoding more samples in each transmission improves the compression ratio, as it exploits the sparsity of the data, yielding higher quality for a given data rate.

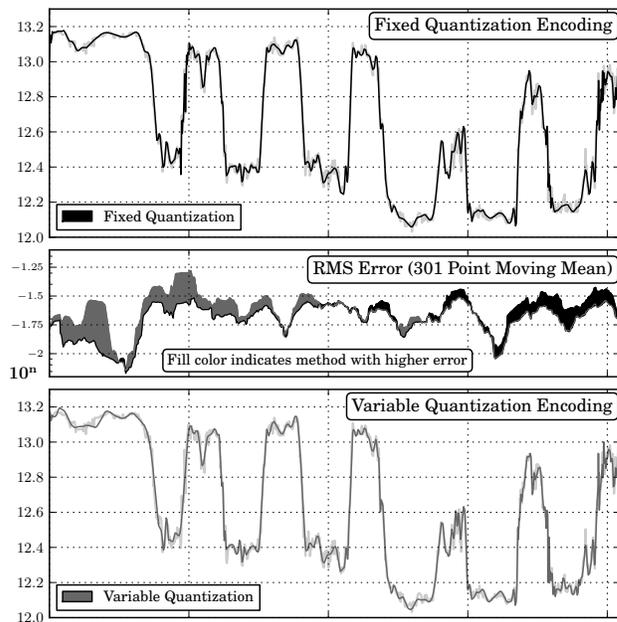


Figure 8: Standard SPIHT compared to our time-varying quantization. Note that time-varying quantization performs better on the more recent data.

gramming language, the case of $n > 1$ is more clearly written as $\text{logspace}(16, 18, n, \text{base} = 2)$.

$$\mathbf{c}_n = \begin{cases} 2^{18} & n = 1 \\ 2^{\{16 + \frac{2x}{n-1} : x \in \mathbb{N}_0, x < n\}} & n > 1 \end{cases} \quad (1)$$

While our cost function was purely time-based, this technique could be used with any function that employs information shared by the encoder and decoder.

4.2 Photos

In February of 2010, we tested these methods during a research expedition aboard the NOAA Ship Oscar Elton Sette, between Guam and the Commonwealth of Northern Mariana Islands. A SeaBED-class AUV was equipped with a WHOI Micro-Modem [6] for communication and navigation, and a five Megapixel high dynamic range Prosilica camera operating at a windowed resolution of 2048x2048 in Bayer RGGB mode.



Figure 9: First six seafloor images returned by AUV.

One background thread of software operation compressed a new image every few minutes, thus ensuring it would not interfere with critical tasks. Prior to compression, a single 2048x2048 RGGB image was converted to a 1024x1024 Y'UV image and separated into color planes. The images

were encoded to a total size of 4032 bytes per image before being segmented into 63-byte “mini-frames” for transmission. During transmission, mini-frames were prefixed with a one byte index identifying the frame offset in the image data, and concatenated into acoustic packets at the native size for the PSK encoding. The surface ship would intermittently report the mini-frames that it had received, which were then removed from queue on the AUV. When all mini-frames for an image had been acknowledged, the AUV began transmission of a newly compressed frame. A total of 15 images were received, 4 of which were completely black as they were captured during descent or ascent. A 16th image was garbled during transmission due to a design error in our (admittedly simple) acknowledgement protocol. The eleven non-black images received are shown in Figures 9 and 10.



Figure 10: The final five non-black seafloor images returned by the AUV during field trials.

The fifteen images were transmitted over a 3.75 hour period, resulting in about fifteen minutes per image. Note that this is largely due to packet loss and scheduling in real world conditions; the PSK encodings that we employed had maximum theoretical burst rates of 520 and 5400 bits per second, compared with the 35 bits per second we achieved. The cover image, Figure 1, illustrates the transmission progress of each image as the vehicle proceeded through its mission.

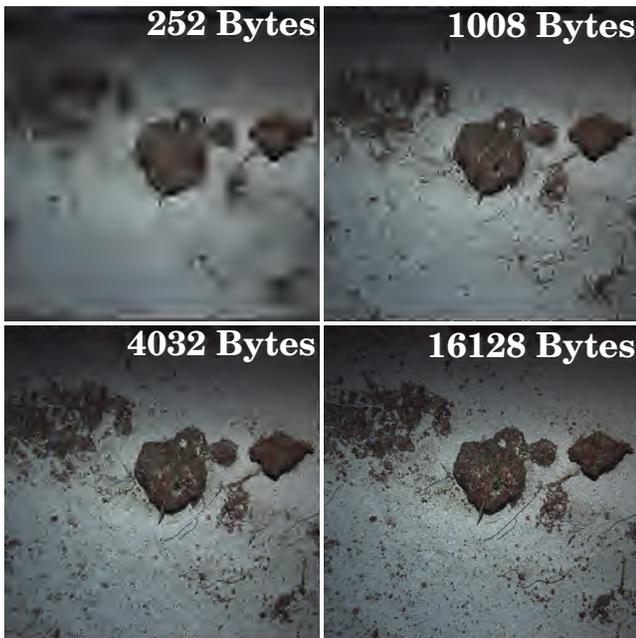


Figure 11: The same image encoded at four different sizes using SPIHT. 81% of the transmitted data is used to reconstruct the luminance, the rest describes the image color channels.

Finally, Figure 11 illustrates a single 1024x1024 pixel color image encoded at four drastically different levels of compression, to give an indication of what transmission at different rates would have looked like.

5. CONCLUSIONS

Previous work has established the utility of the DWT transform in oceanographic data compression and analysis, but there has been more limited coverage of complete compression solutions for oceanographic data. Set partitioning methods, such as those based upon SPIHT and presented here, provide a simple and promising method for performing compression on a variety of one dimensional numerical data, and real-world images. Even at extremely high compression ratios, SPIHT provides human-interpretable data to surface operators.

The use of a fully embedded representation for science telemetry enables recipients to request additional detail for areas of interest, whether the recipient is a human on the surface or an algorithm running on a different network node. As AUV technology advances towards truly autonomous decision-making, providing surface 'supervisors' with data to complement these decisions will open new doors.

6. ACKNOWLEDGMENTS

We would like to thank the NSF Gordon-CENSSIS ERC, who funded this research under grant EEC-9986821, and NOAA's Pacific Islands and Northwest Fisheries Science Centers for their continued support and close collaboration. Finally, thanks to Commander Anita L. Lopez and the crew of the Oscar Elton Sette for their extraordinary support during field trials.

7. REFERENCES

- [1] I. F. Akyildiz, D. Pompili, and T. Melodia. Underwater Acoustic Sensor Networks: Research Challenges. In *Ad Hoc Networks*, volume 3, pages 257 – 279. Elsevier, Mar. 2005.
- [2] M. Burtscher and P. Ratanaworabhan. High throughput compression of double-precision floating-point data. In *2007 Data Compression Conference (DCC'07)*, 2007.
- [3] R. A. DeVore and B. J. Lucier. Wavelets. In *Acta Numerica 1*, pages 1–56. Cambridge University Press, 1992.
- [4] D. L. Donoho, M. Vetterli, R. DeVore, and I. Daubechies. Data Compression and Harmonic Analysis. *IEEE Transactions on Information Theory*, 44(6), Oct. 1998.
- [5] R. L. Eastwood, L. E. Freitag, and J. A. Catipovic. Compression techniques for improving underwater acoustic transmission of images and data. In *Proceedings of MTS/IEEE OCEANS 1996: Prospects for the 21st Century*, Fort Lauderdale, FL, Sept. 1996.
- [6] L. Freitag, M. Grund, S. Singh, J. Partan, P. Koski, and K. Ball. The WHOI Micro-Modem: An Acoustic Communications and Navigation System for Multiple Platforms. In *Oceans 2005, Proceedings of the MTS/IEEE*, volume 2, pages 1086 – 1092, 2005.
- [7] T. W. Fry and S. Hauck. SPIHT image compression on FPGAs. *Circuits and Systems for Video Technology, IEEE Transactions on*, 15(9):1138–1147, sep 2005.
- [8] J. R. Goldschneider. *Lossy Compression of Scientific Data via Wavelets and Vector Quantization*. PhD thesis, University of Washington, Seattle, WA, June 1997.
- [9] D. F. Hoag and V. K. Ingle. Underwater image compression using the wavelet transform. In *OCEANS '94. Proceedings*, volume 2, pages 533–537, Sep 1994.
- [10] D. F. Hoag, V. K. Ingle, and R. J. Gaudette. Low-bit-rate coding of underwater video using wavelet-based compression algorithms. *IEEE Journal of Oceanic Engineering*, 22(2):393–400, Apr. 1997.
- [11] M. Jakuba, C. Roman, H. Singh, C. Murphy, C. Kunz, C. Willis, T. Sato, and R. Sohn. Field report: Long-baseline acoustic navigation for under-ice AUV operations. *J. Field Robotics*, 25(8), July 2008.
- [12] C. Kunz, C. Murphy, H. Singh, C. Willis, R. A. Sandip Singh, T. Sato, K. Nakamura, M. Jakuba, R. Eustice, R. Camilli, and J. Bailey. Toward extraplanetary under-ice exploration: Robotic steps in the arctic. *Journal of Field Robotics Special Issue on Space Robotics*, submitted.
- [13] P. Lindstrom and M. Isenburg. Fast and efficient compression of floating-point data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5), Sept. 2006.
- [14] W. A. Pearlman and A. Said. Set partition coding: Part I of set partition coding and image wavelet coding systems. In *Foundations and Trends in Signal Processing*, pages 95–180. Now Publishers, Jun 2008.
- [15] C. Pelekanakis, M. Stojanovic, and L. Freitag. High rate acoustic link for underwater video transmission. *OCEANS 2003. Proceedings*, 2:1091–1097, Sep 2003.
- [16] S. Saha. Image Compression - from DCT to Wavelets :

- A Review. *ACM Crossroads: Data Compression*, Spring 2000.
- [17] A. Said and W. A. Pearlman. A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *Circuits and Systems for Video Technology, IEEE Transactions on*, 6(3):243–250, Jun 1996.
- [18] D. Salomon. *Data Compression: The Complete Reference*, chapter 4.5. Springer, 4 edition, 2007.
- [19] C. Sayers, A. Lai, and R. Paul. Visual imagery for subsea teleprogramming. In *Proc. IEEE Robotics and Automation Conference*, 2005.
- [20] T. Schneider and H. Schmidt. The dynamic compact control language: A compact marshalling scheme for acoustic communications. In *Proceedings of MTS/IEEE OCEANS 2010*, Sydney, Australia, 2010.
- [21] J. M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *Signal Processing, IEEE Transactions on*, 41(12):3445–3462, Dec 1993.
- [22] H. Singh, A. Can, R. Eustice, S. Lerner, N. McPhee, O. Pizarro, and C. Roman. SeaBED AUV offers new platform for high-resolution imaging. *EOS, Transactions of the AGU*, 85(31):294–295, Aug. 2004.
- [23] M. Stojanović. Recent advances in high-speed underwater acoustic communications. *IEEE Journal of Oceanic Engineering*, 21(2):125–136, Apr. 1996.
- [24] R. Stokey, L. Freitag, and M. Grund. A Compact Control Language for auv acoustic communication. *Oceans 2005 - Europe*, 2:1133–1137, June 2005.
- [25] M. Suzuki, T. Sasaki, and T. Tsuchiya. Digital acoustic image transmission system for deep-sea research submersible. *Oceans 1992, Proceedings of the MTS/IEEE*, 2:567–570, Oct 1992.
- [26] M. Vetterli. Wavelets, approximation, and compression. *IEEE Signal Processing Magazine*, pages 59 – 73, Sept. 2001.
- [27] T. A. Welch. A technique for high-performance data compression. *Computer*, 17(6):8–19, June 1984.
- [28] D. R. Yoerger, M. V. Jakuba, A. M. Bradley, and B. Bingham. Techniques for deep sea near bottom survey using an autonomous vehicle. *International Journal of Robotics Research*, 26(1):41–54, 2007.
- [29] J. Ziv and A. Lempel. A universal algorithm for data compression. *IEEE Transactions on Information Theory*, 23(3):337–343, May 1977.