# Wavelet Compression with Set Partitioning for Low Bandwidth Telemetry from AUVs


Chris Murphy

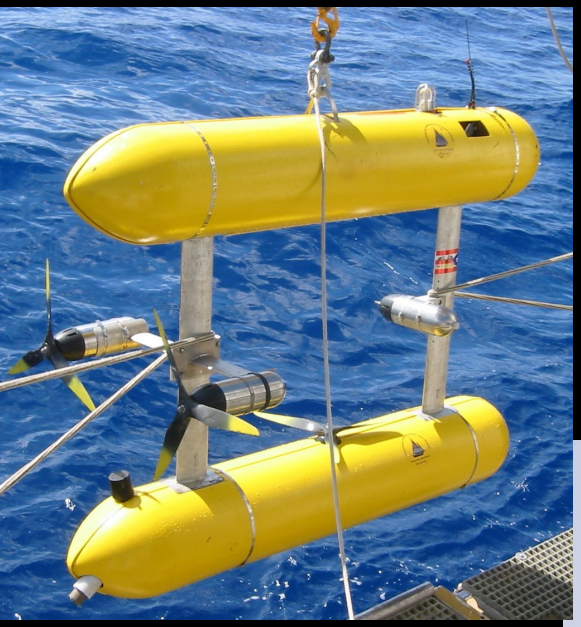
Dr. Hanumant Singh

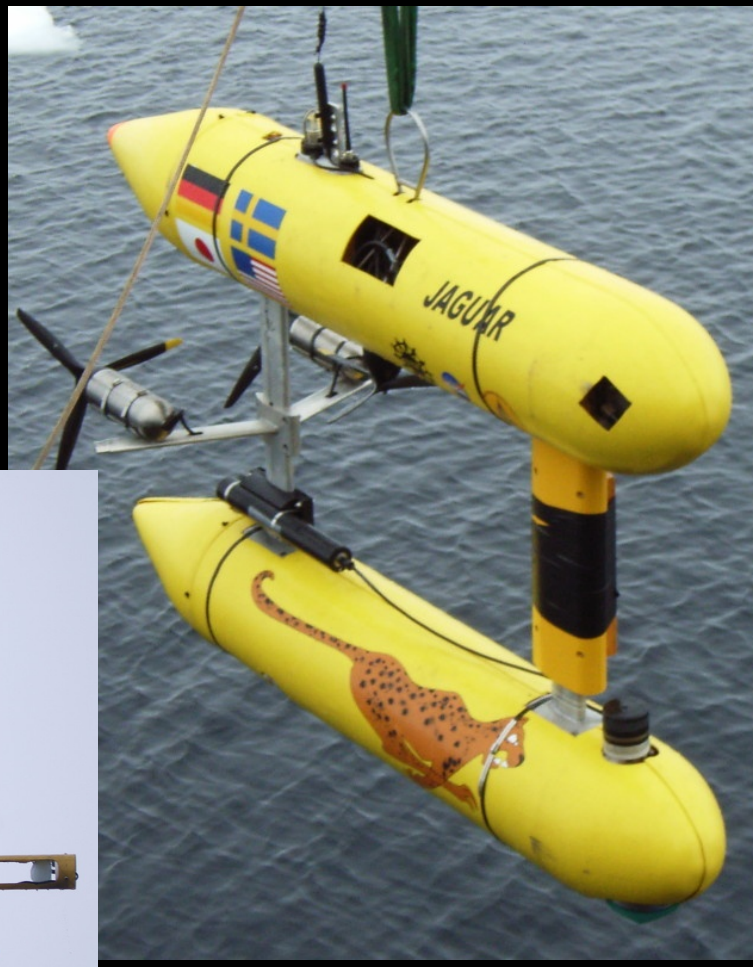
WOODS HOLE OCEANOGRAPHIC INSTITUTION
1930

WUWNet, 30 September 2010

# The SeaBED Family

**933 MHz x86 CPU**
**Ubuntu Linux**
**C / C++ / Perl / Python**
**10kHz MicroModem + PSK**
**2km or 6km depth**
**Cameras + Multibeam**
**Flexible sensor package**




WOODS HOLE OCEANOGRAPHIC INSTITUTION
1930

# WHOI Acoustic MicroModem

## Three Acoustic Encoding Methods

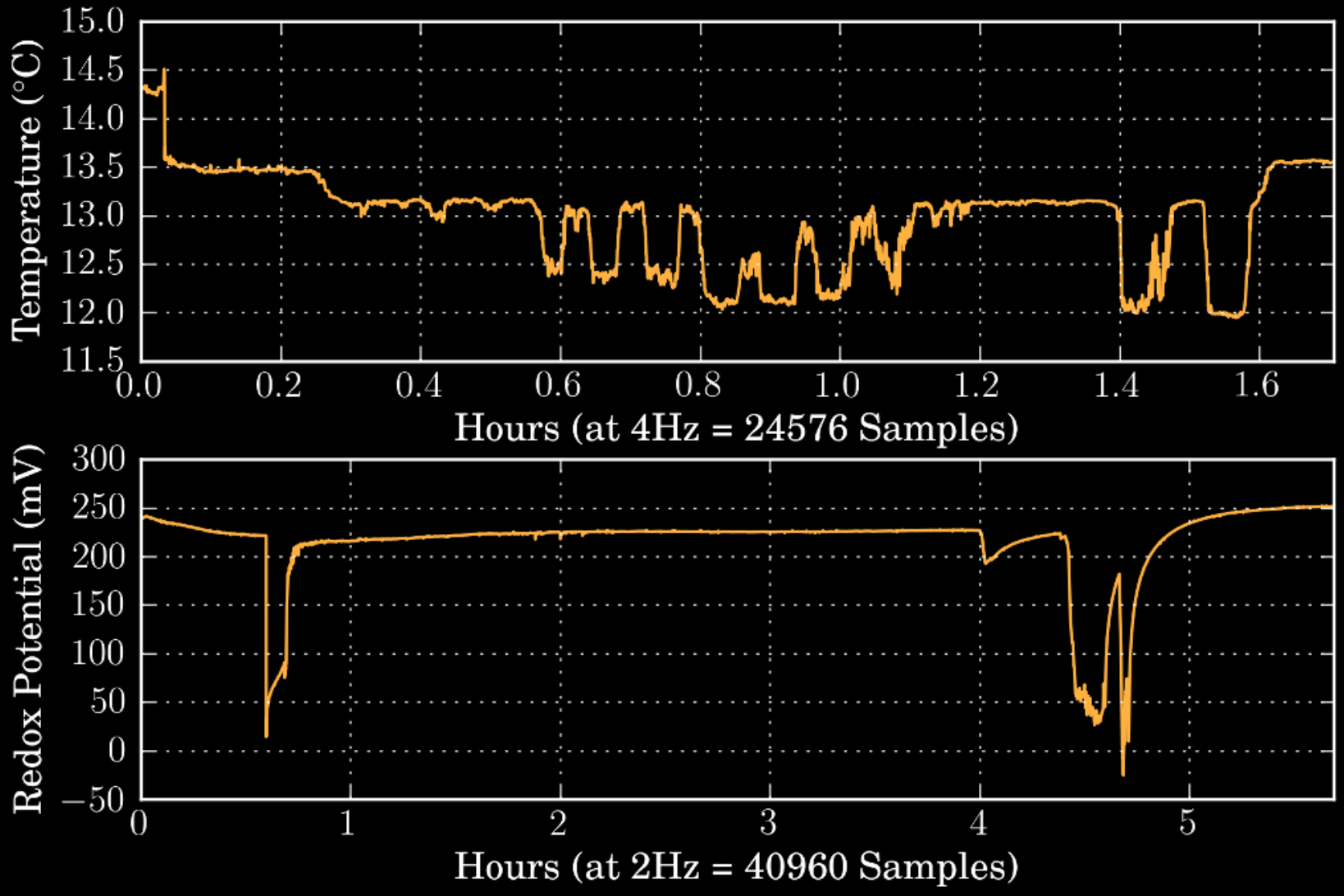
| FH-FSK | 1/15 Spreading PSK | 9/14 Rate Block Code PSK |
|---|---|---|
| 10Bps Maximum Rate<br><br>Each transmission is:<br>1 Frame @ 32 Bytes | 62Bps Maximum Rate<br><br>Each transmission is:<br>3 Frames @ 64 Bytes | 662Bps Maximum Rate<br><br>Each transmission is:<br>8 Frames @ 256 Bytes |

Ongoing usage for vehicle health telemetry.

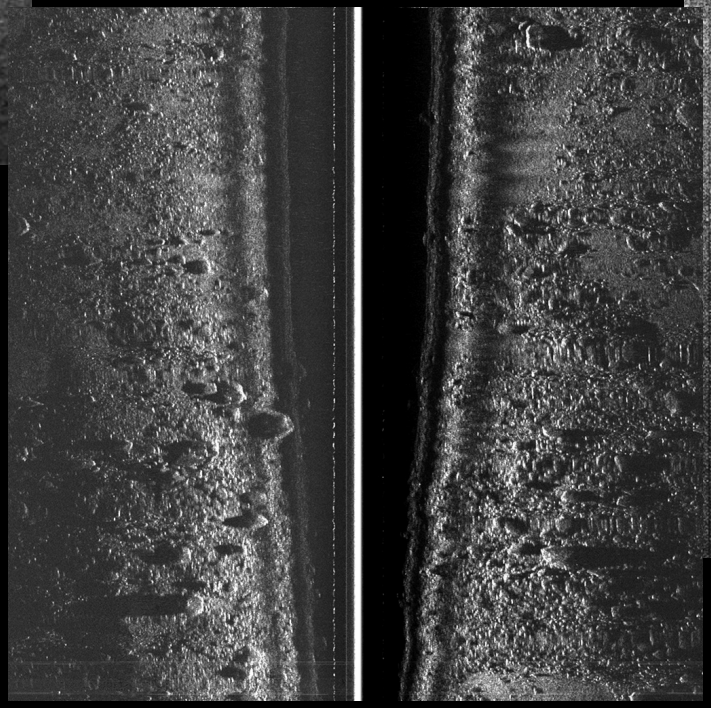
Used for higher-rate transmission of compressed data described here.

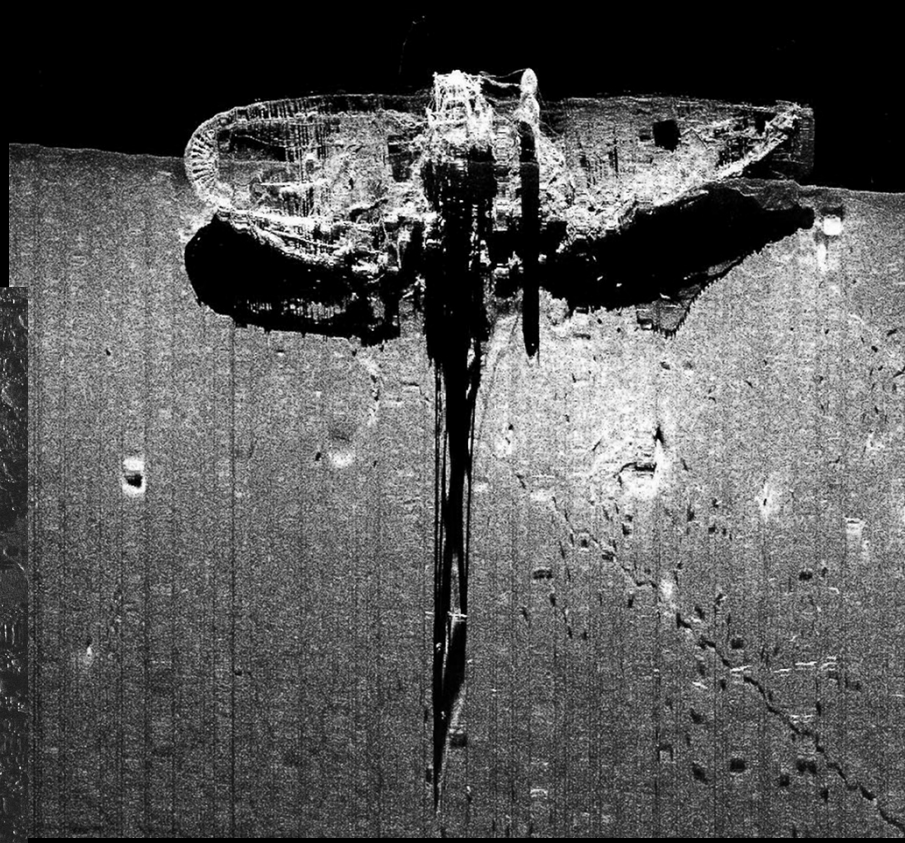
Requires DSP co-processor on receiver side.

# Environmental Sensors



# Sidescan Sonar Imagery

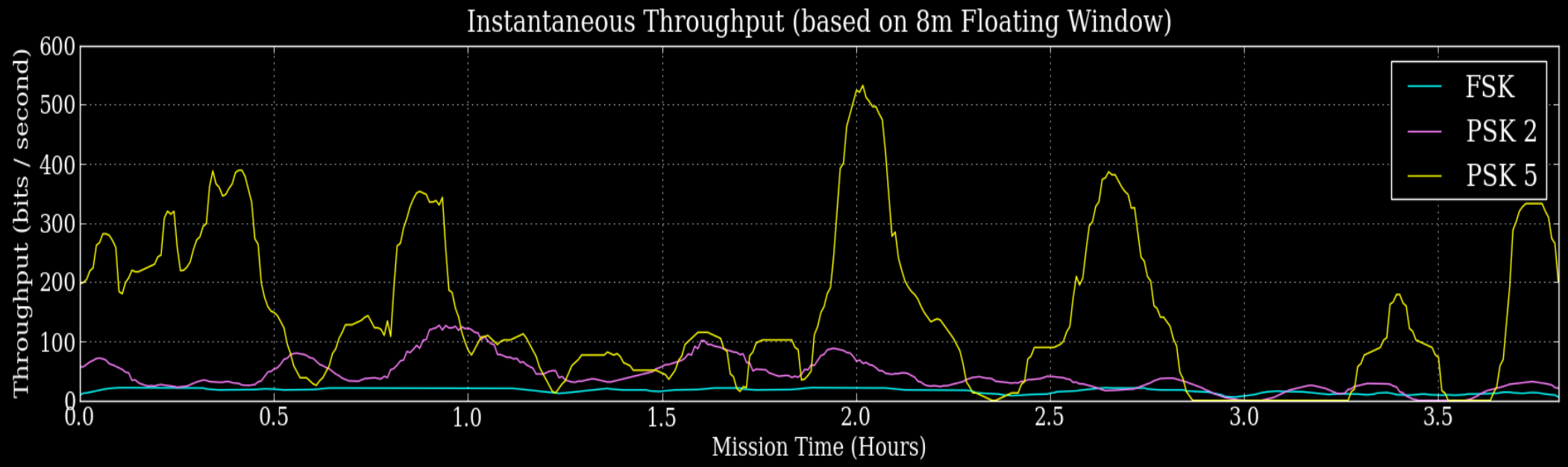




# Photographs

# Low Throughput

**In real-world conditions, effective throughput of acoustic modem can be 10-100 bits per second (or worse).**



# The Problem

**Data is typically unavailable until after recovery.**

**Effective collaboration requires information sharing, whether from sub-sea to surface or sub-sea to sub-sea.**

**How can underwater vehicles effectively share multimodal information at these rates?**

# Packetized Communications

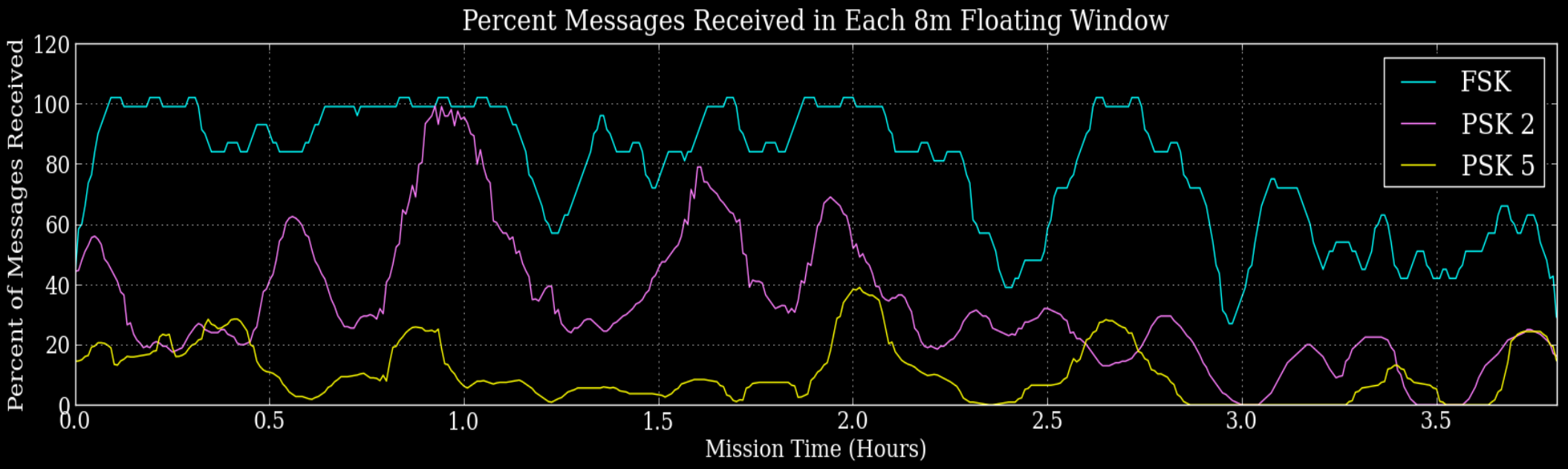
**We have historically used CCL and similar methods, making each acoustic packet fully self contained.**

CCL 'scixy_owtt' Frame

| Byte | Description |
|---|---|
| 0 | Type: CCL_SCIXY_OWTT |
| 1<br>2<br>3 | X position in Meters |
| 4<br>5<br>6 | Y position in Meters |
| 7 | Heading in $\frac{360}{255}$ths of a degree |
| 8<br>9 | Depth in Meters |
| 10<br>11 | Altitude in Meters |
| 12<br>13 | Goal ID |
| 14<br>15<br>16 | Goal X position in Meters |
| 17<br>18<br>19 | Goal Y position in Meters |
| 20<br>21 | Goal Depth in Meters |
| 22<br>23 | LBL#3 Travel Time in Sec. |
| 24<br>25 | LBL#4 Travel Time in Sec. |
| 26<br>27<br>28<br>29<br>30 | Arbitrary Science Payload |
| 31 | One-way Travel Time data |

# Why?

**Surface operators get antsy if they don't know what the AUV is doing. Even when it's working fine.**



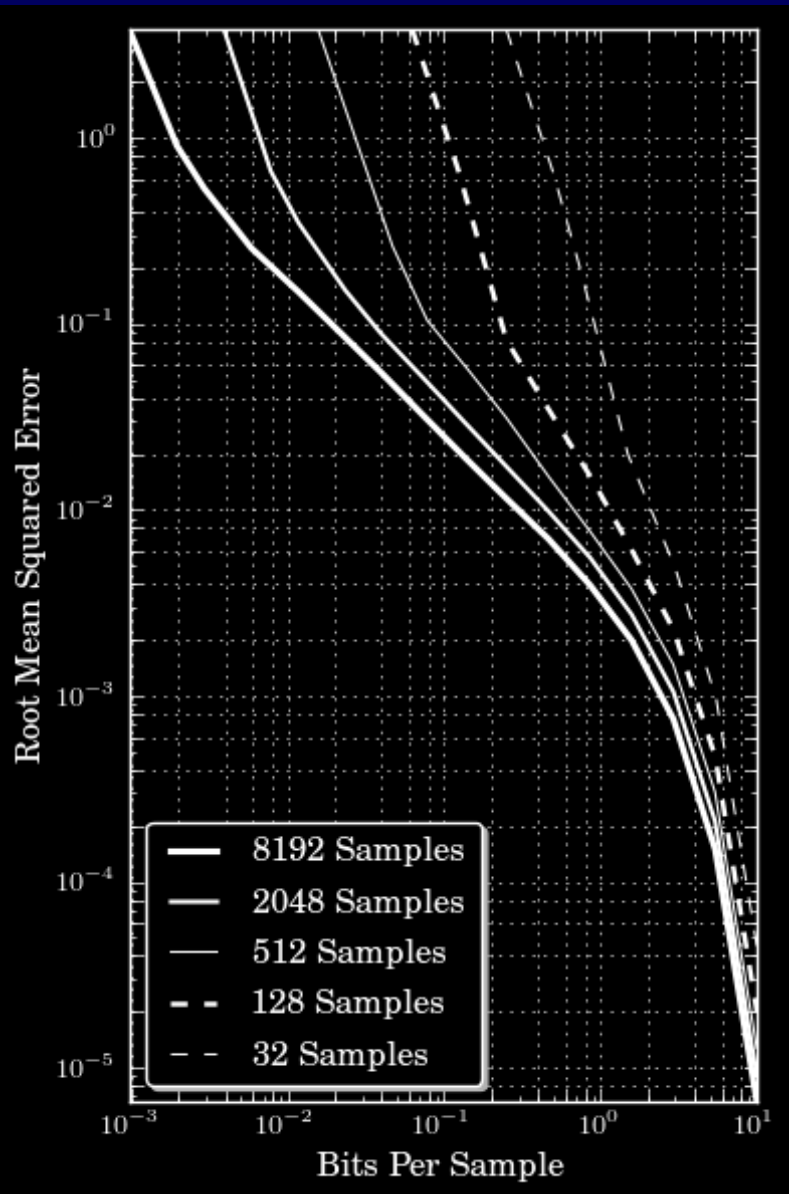

**As vehicle reliability increases, this is changing.**

# Transform Compression

**Transform compression employs inherent data sparsity in some transformed basis.**

**It is easier to compress *more* (correlated) data than less.**

**Thus, <u>the less frequently you share information the more efficiently you can share it</u>.**



# Fully Embedded Coding

**Fully embedded coding –** A high quality version of encoded data shares first N bits *identically* with a poor quality encoding of length N.


If preview is interesting, recipients can request additional packets to obtain high quality sections of signals or images.

# SPIHT

## Set Partitioning In Hierarchical Trees *

**Operates in Wavelet Domain**

- Wavelets have been shown to provide an efficient, sparse representation for a variety of real-world signals.

**Dimensionality-Independent**

- Same encoding works for CTD data, Imagery, Volumetric data

**Embedded Coding**

- Low-fidelity versions are *identical* to the beginning of high-fidelity
- Each additional (in-order) byte improves the estimate
- Sending up a higher quality version doesn't require 'starting over'

** Said and Pearlman, 1996*

# Brief Introduction to SPIHT

Step 1: Transform signal with Discrete Wavelet Transform.

```
threshold = get_starting_threshold()
while threshold > 0:
   sort(threshold)
   refine(threshold)
   threshold >>= 1
```

Step 2: iterates over a decreasing threshold value, identifying "significant" wavelet coefficients larger than the threshold.

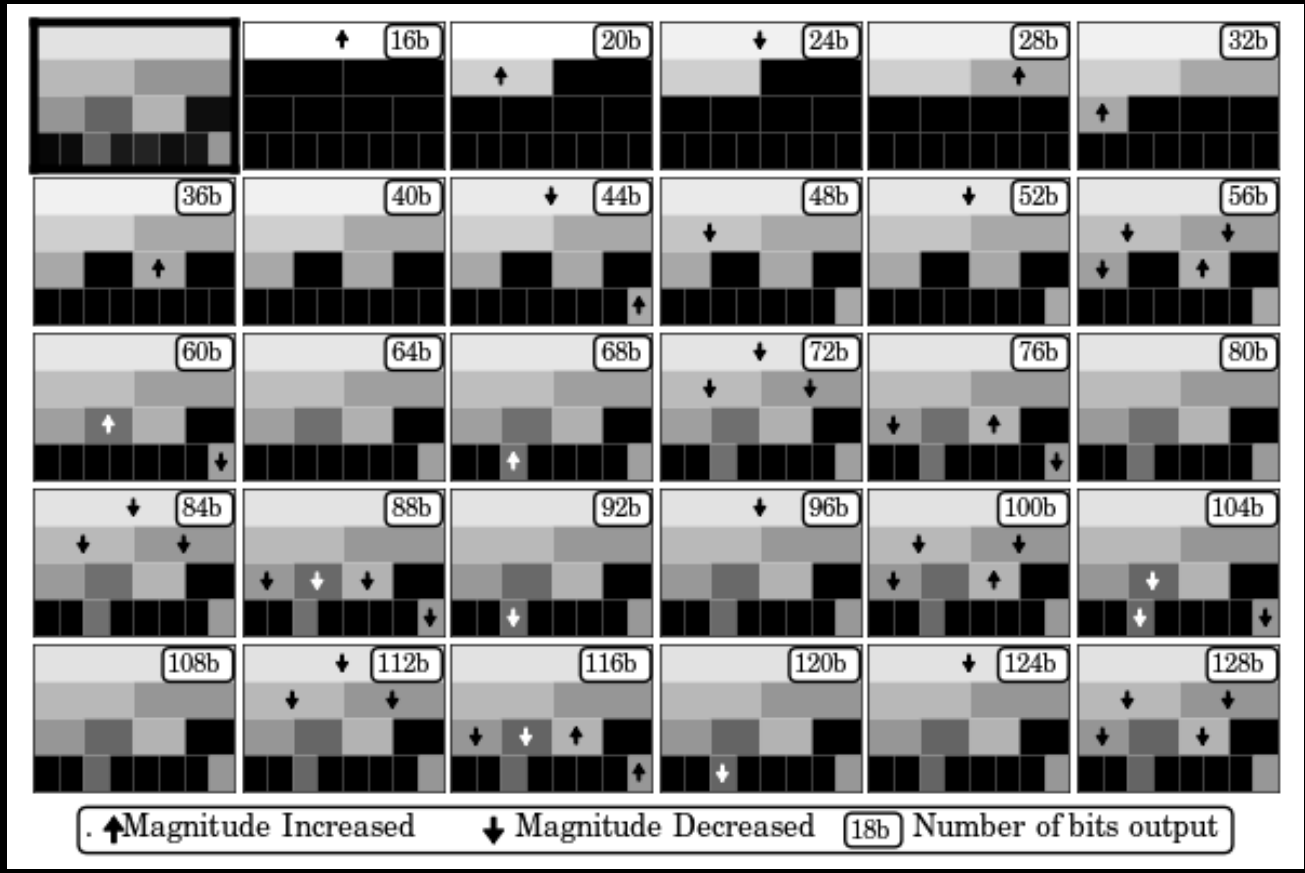
# Brief Introduction to SPIHT

**Sorting Bits Indicate**

- Is a coefficient "significant"?
- Are any descendants?
- Any grand-descendant?

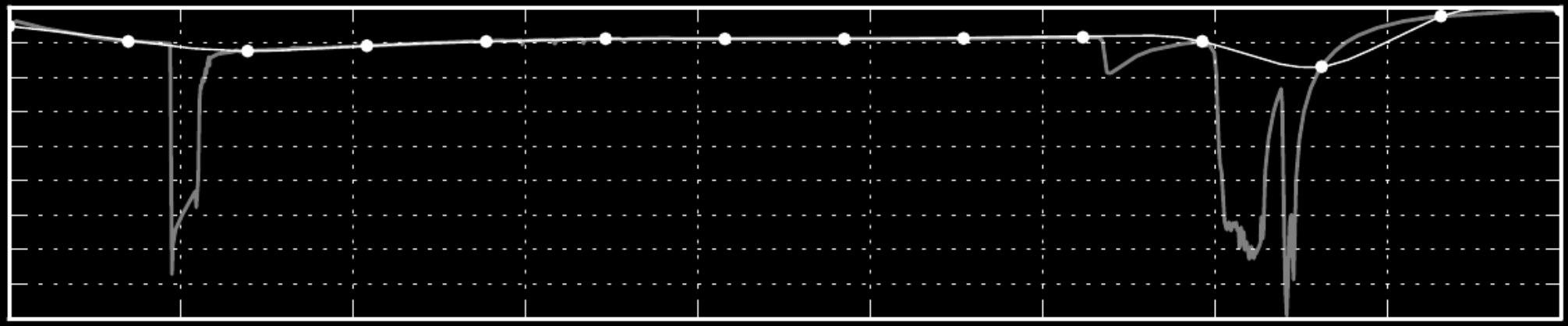
**Refinement Bits Indicate**

- The sign of a coefficient
- Coefficient magnitude

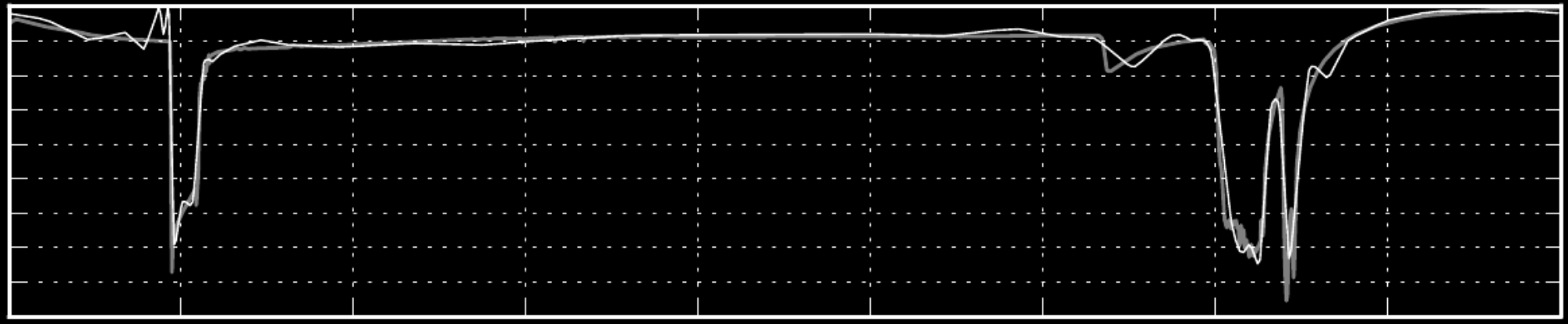


# Reduction Potential - 28 Bytes

14 Spline-Interpolated 16-bit Fixed Point Samples

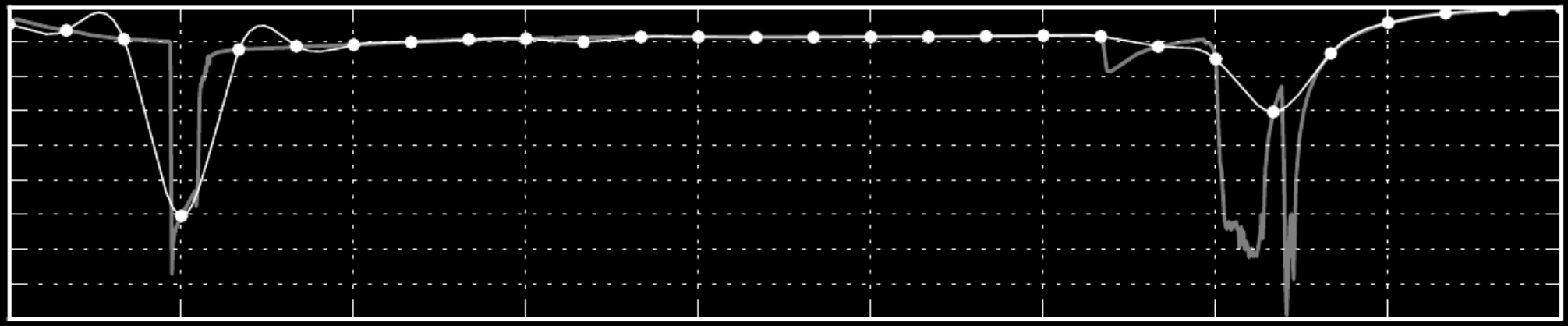


SPIHT Encoded with 28 Bytes

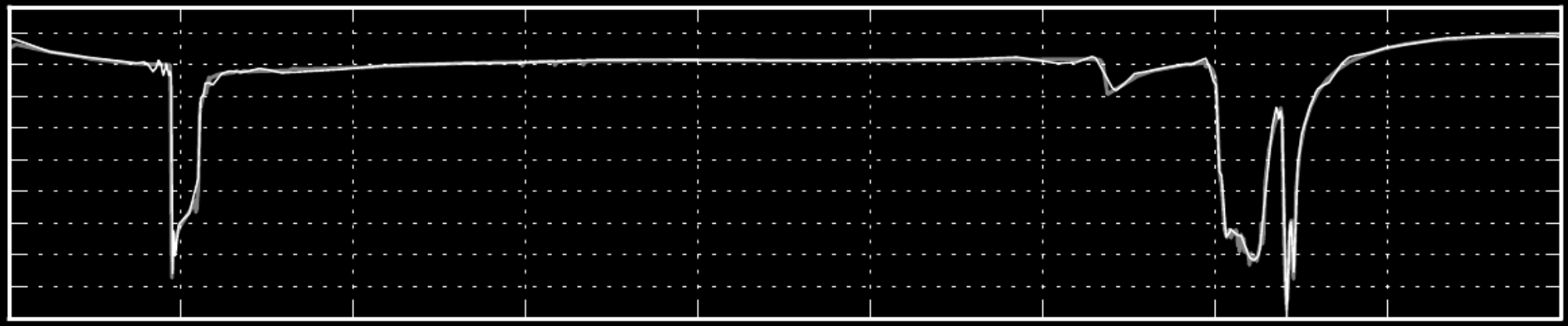


# Reduction Potential - 56 Bytes

28 Spline-Interpolated 16-bit Fixed Point Samples

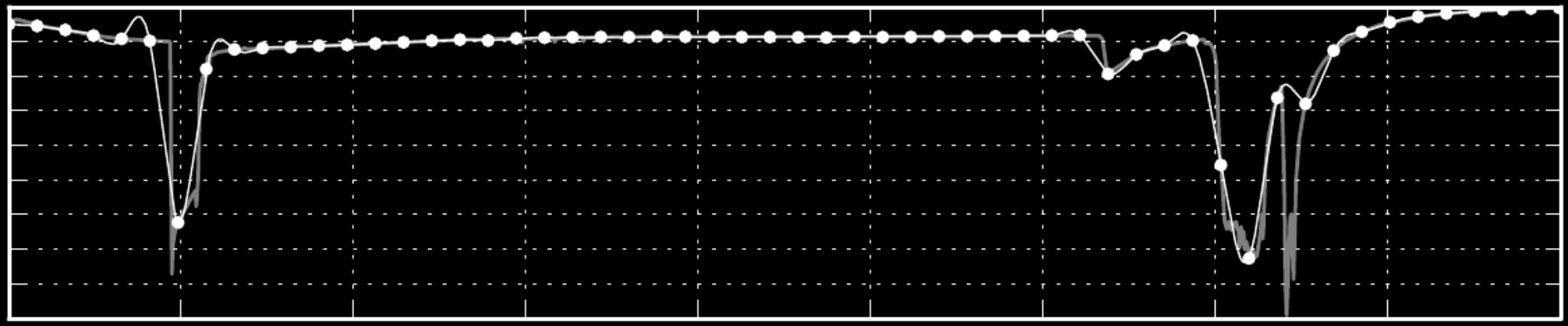


SPIHT Encoded with 56 Bytes

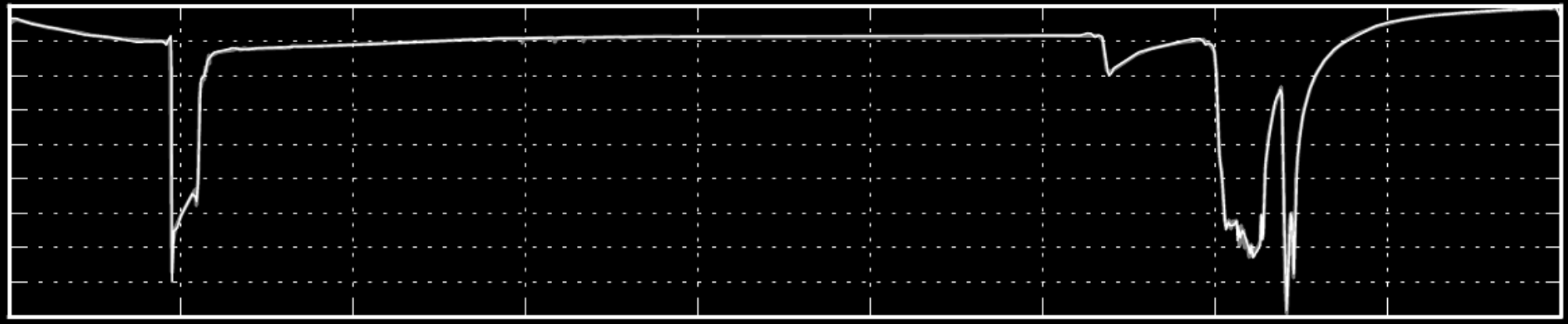


# Reduction Potential - 112 Bytes

56 Spline-Interpolated 16-bit Fixed Point Samples

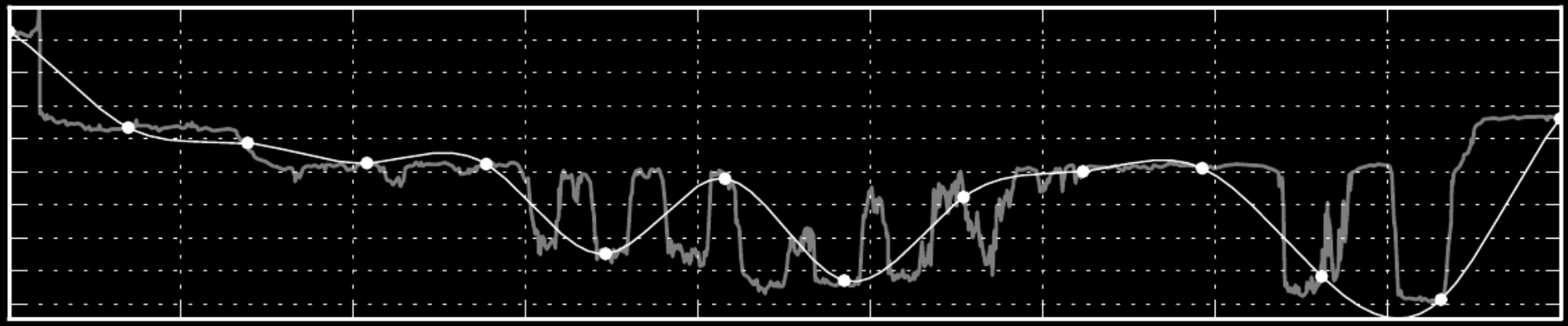


SPIHT Encoded with 112 Bytes

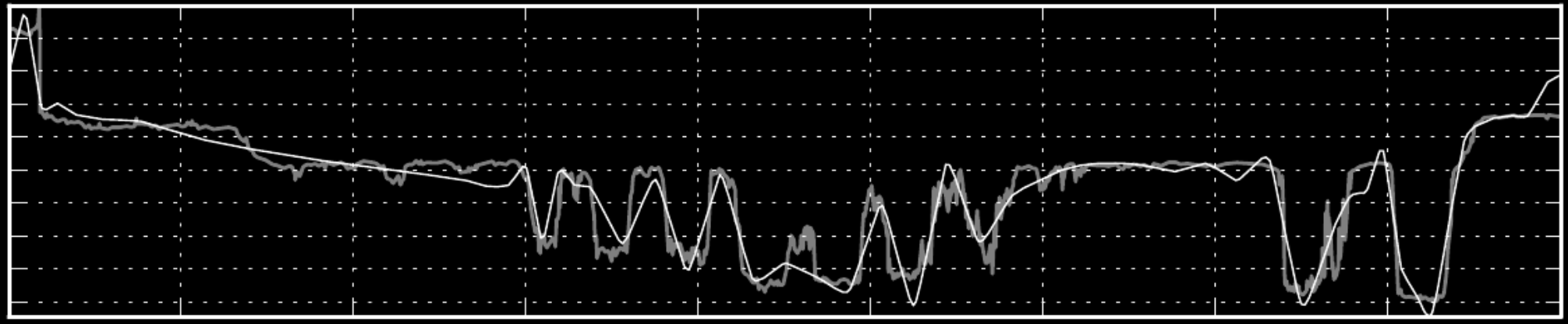


# Temperature - 28 Bytes

14 Spline-Interpolated 16-bit Fixed Point Samples

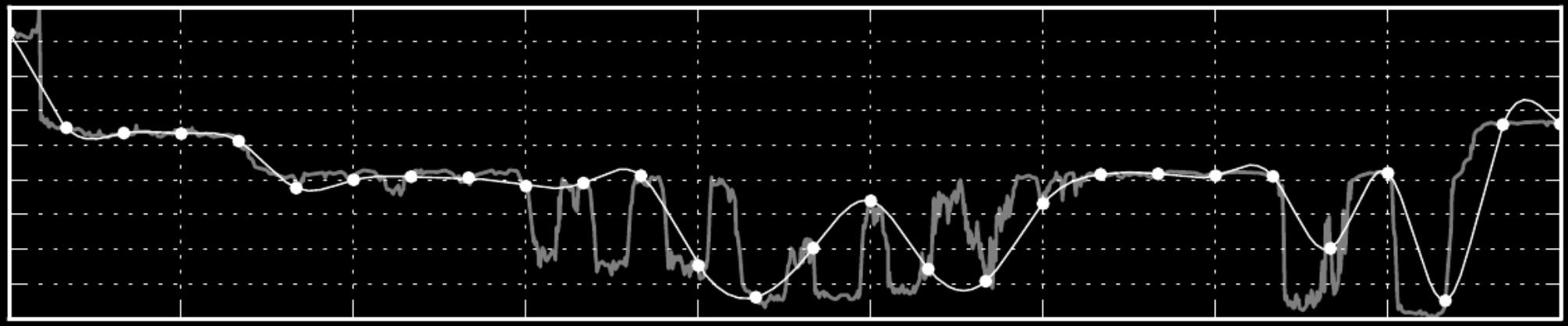


SPIHT Encoded with 28 Bytes

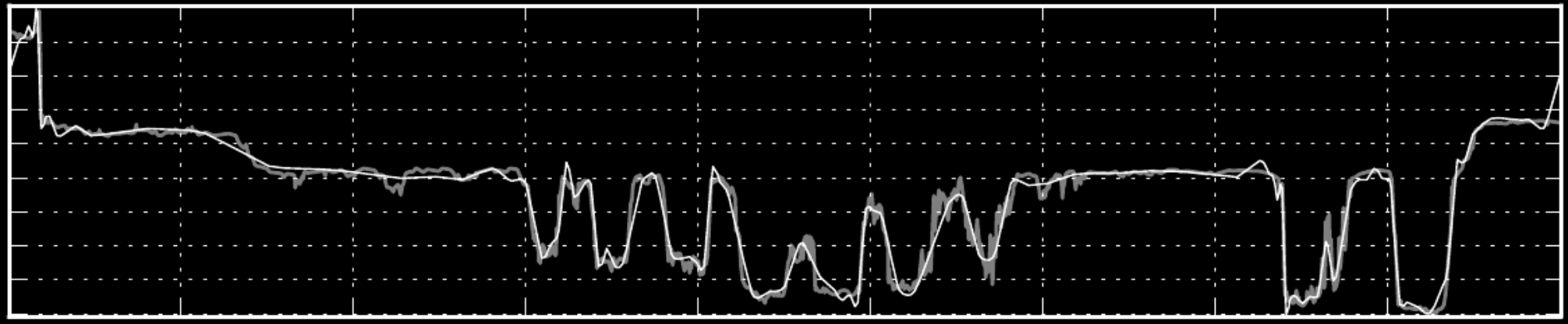


# Temperature - 56 Bytes

28 Spline-Interpolated 16-bit Fixed Point Samples

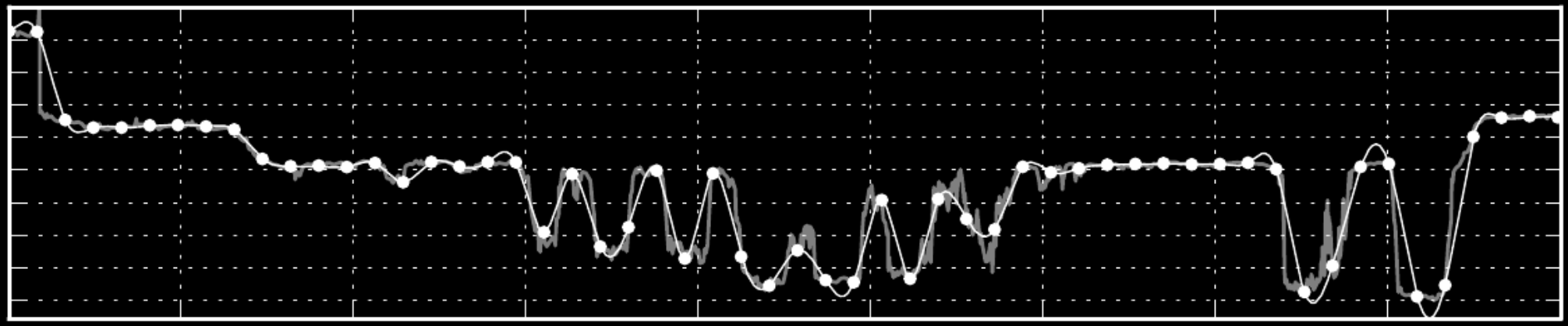


SPIHT Encoded with 56 Bytes

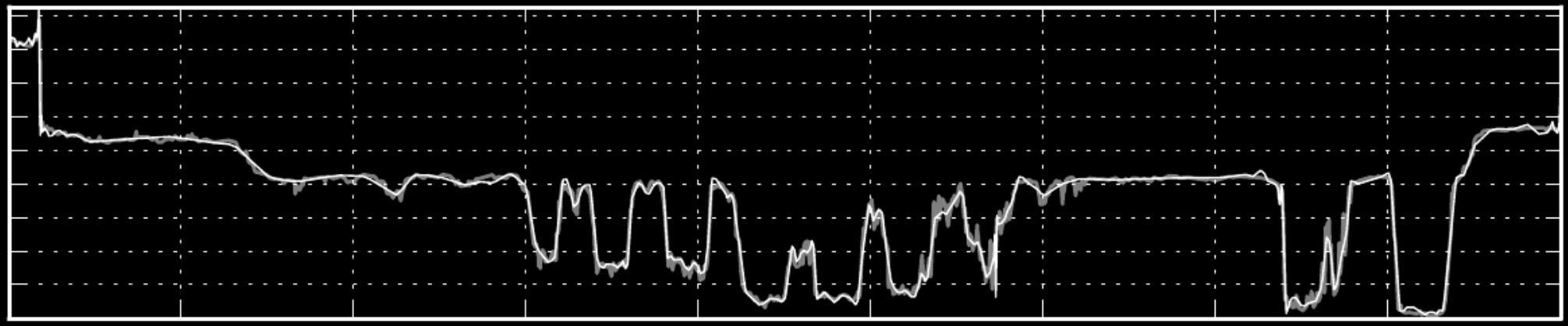


# Temperature - 112 Bytes

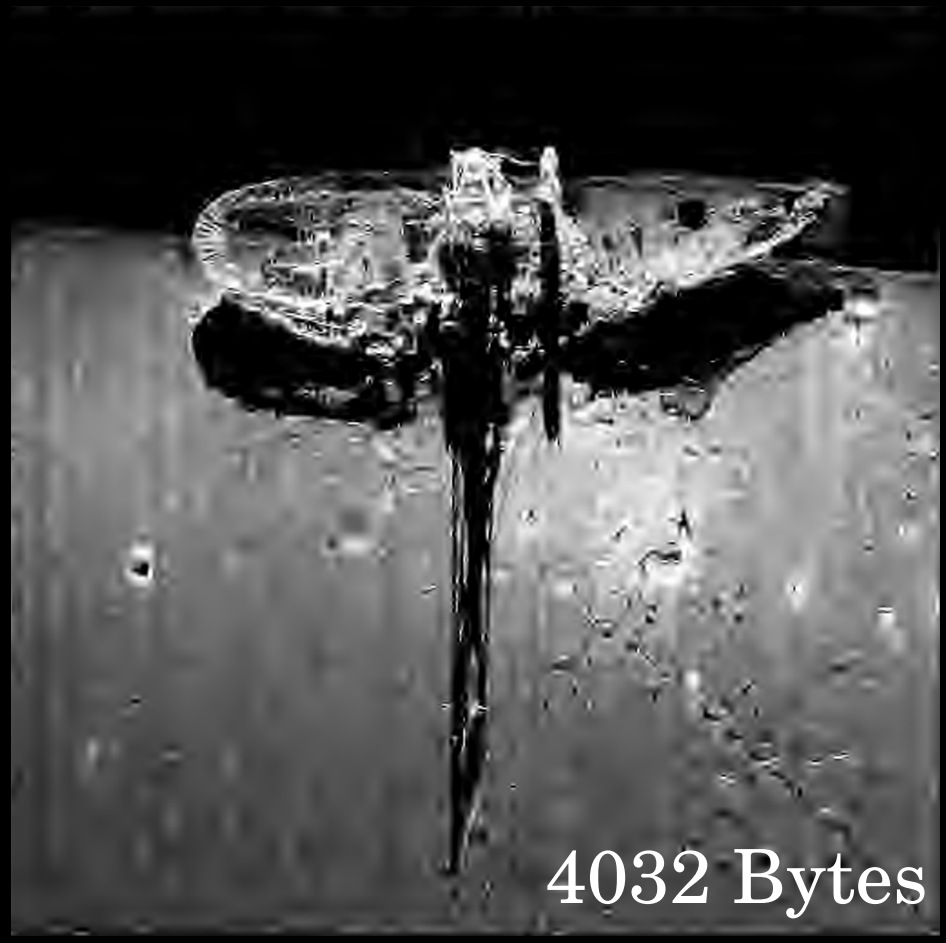
56 Spline-Interpolated 16-bit Fixed Point Samples



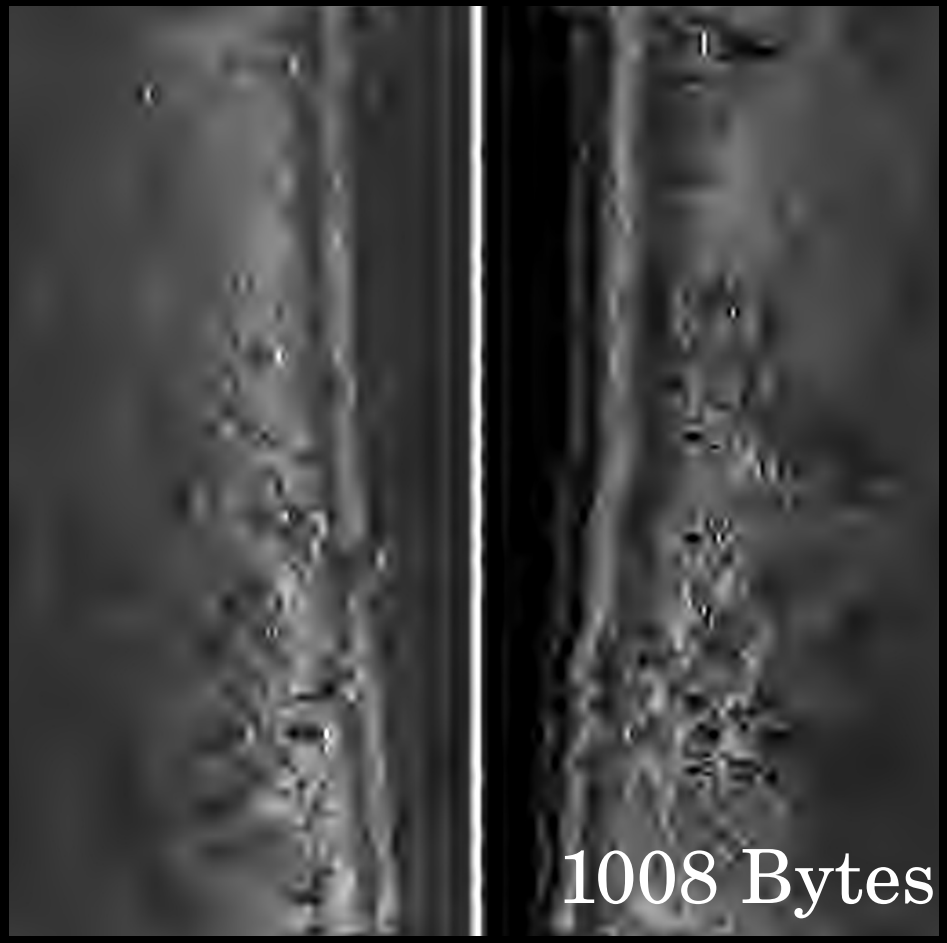
SPIHT Encoded with 112 Bytes

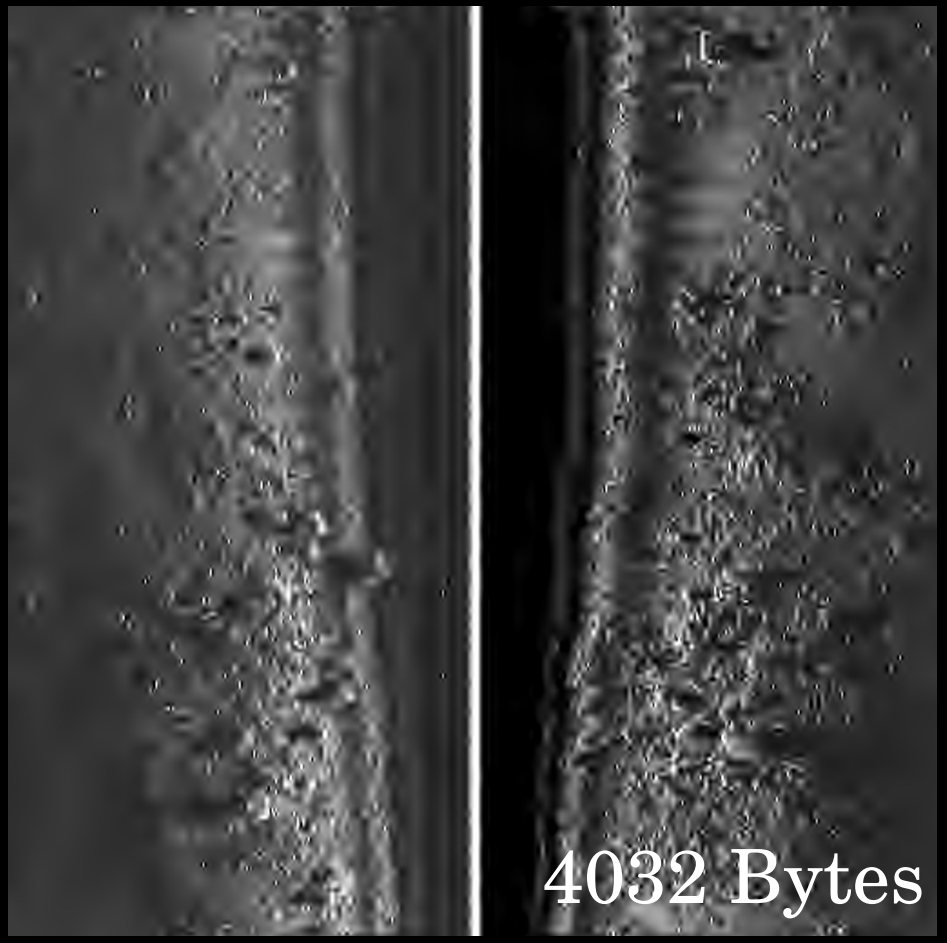


# Sidescan Imagery

# Sidescan Imagery



# Sidescan Imagery

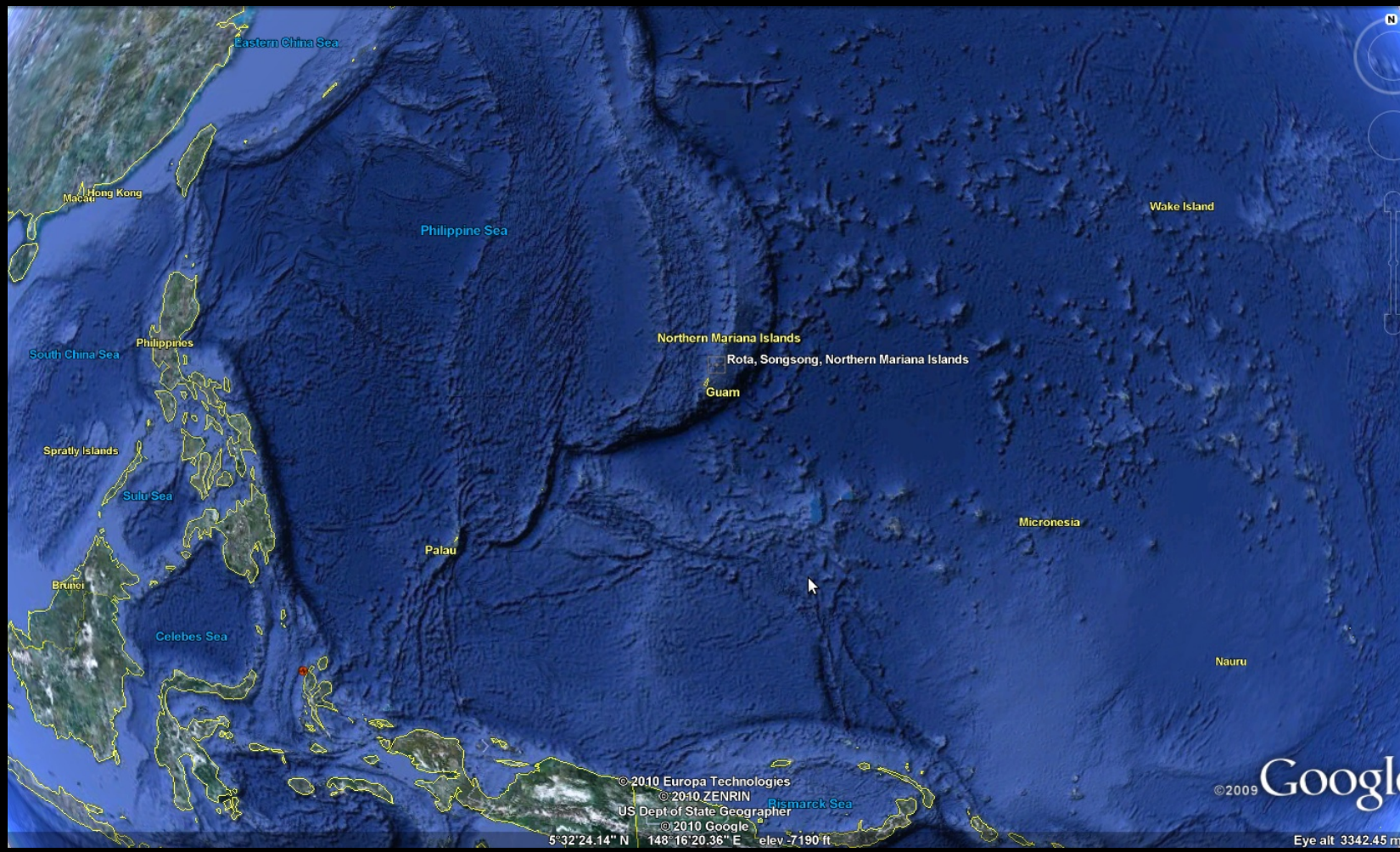

# Live Trials near Rota

North of Guam, 1600mi east of the Philippines

**Depth**
100m → 350m

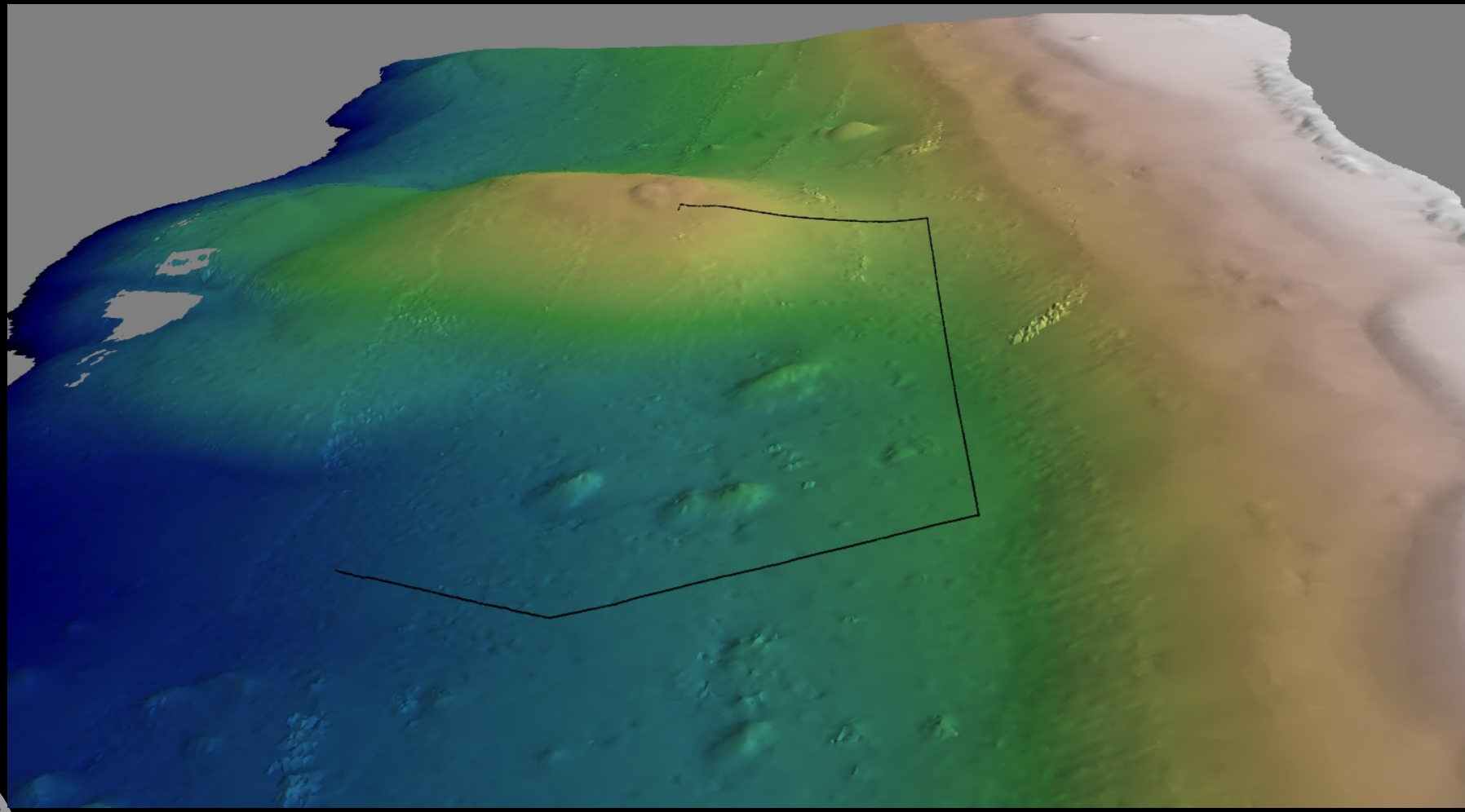
**Ship Range**
100's of meters

**Transducer**
Single ITC
Hemispherical



# Mission Outline

**Land on a small plateau and work down, observing fisheries health.**



# System Overview

**SeaBED + Receiver Code**

**SPIHT**

**Ad-Hoc Segmentation Implementation**

**Implemented by WHOI MicroModem**

| OSI Model | | | |
|---|---|---|---|
| | **Data unit** | **Layer** | **Function** |
| | | 7. Application | Network process to application |
| **Host layers** | Data | 6. Presentation | Data representation,encryption and decryption,convert machine dependent data to machine independent data |
| | | 5. Session | Interhost communication |
| | Segments | 4. Transport | End-to-end connections and reliability,Flow control |
| **Media layers** | Packet | 3. Network | Path determination and logical addressing |
| | Frame | 2. Data Link | Physical addressing |
| | Bit | 1. Physical | Media, signal and binary transmission |

**From Wikipedia: OSI model**

# System Implementation

## SeaBED Code

**SeaBED vehicle capturing image every 4 seconds**

- → **2048 x 2048 – four megapixel image**
- → **Bayer-coded / “Raw” image**

**Separate encoder thread**

- → **Rate limited – sleeps for a few minutes after encoding**
- → **Downsamples RAW image to 1024 x 1024 RGB image**
- → **RGB converted to Y'UV colorspace**
- → **Y Channel encoded to fixed size of 2016 Bytes**
- → **U,V Channels encoded to fixed size of 1008 Bytes**

# System Implementation

## Ad-Hoc Segmentation Implementation

**4032 Byte Image segmented into sixty-four, 63-Byte fragments**

**Single byte "offset" index prepended to fragment**

**64-Byte fragments packed into modem frames, and transmitted**

**uM Rate 2: 64B Frames** **uM Rate 5: 256B Frames**

# System Implementation

## Receiver Code

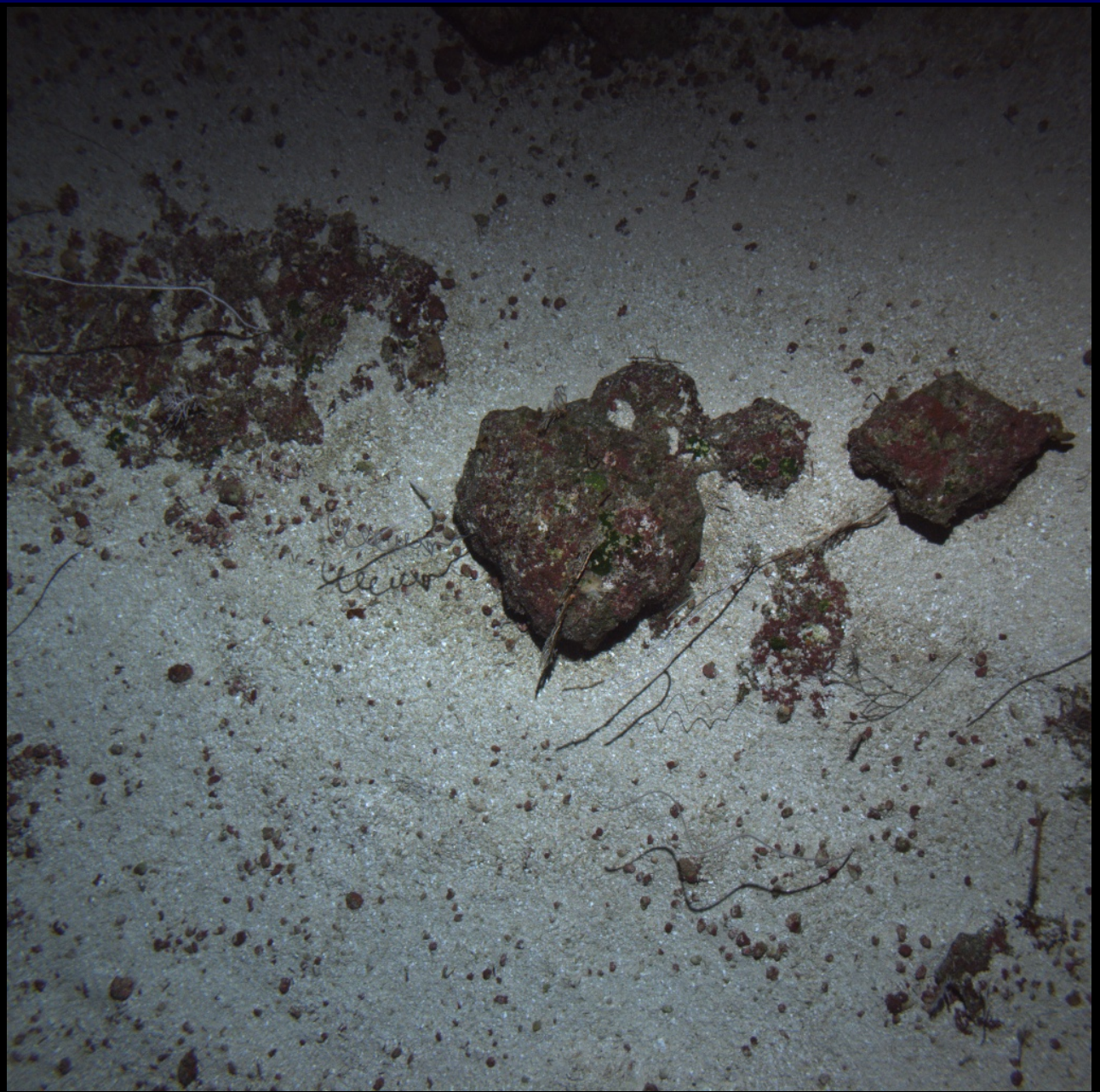
**Fragments written to file, at offset, as they are received**

**No attempt to display partial transmissions was made for this test**

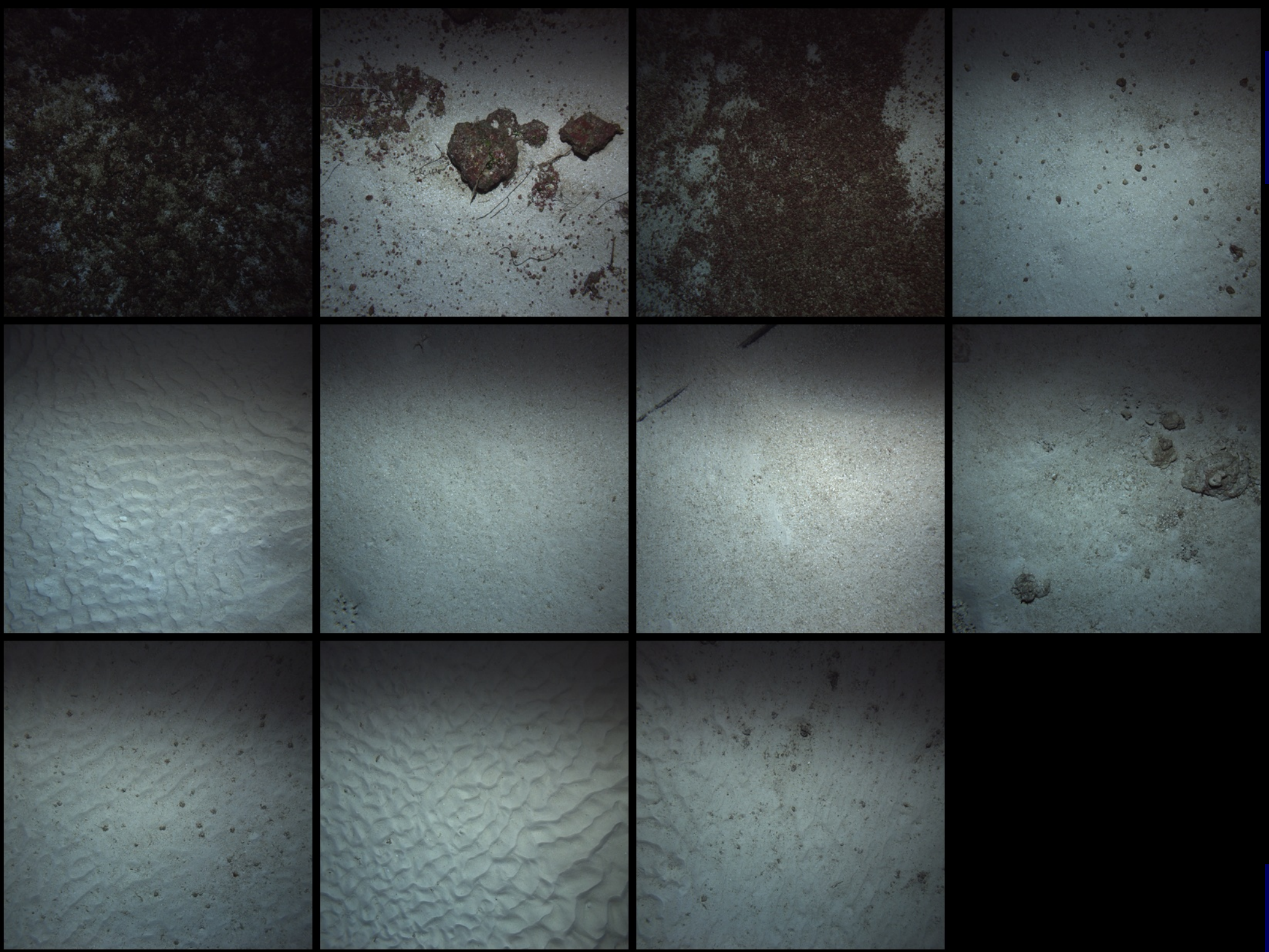
**ACK's were hand-crafted and transmitted (list of offset IDs)**

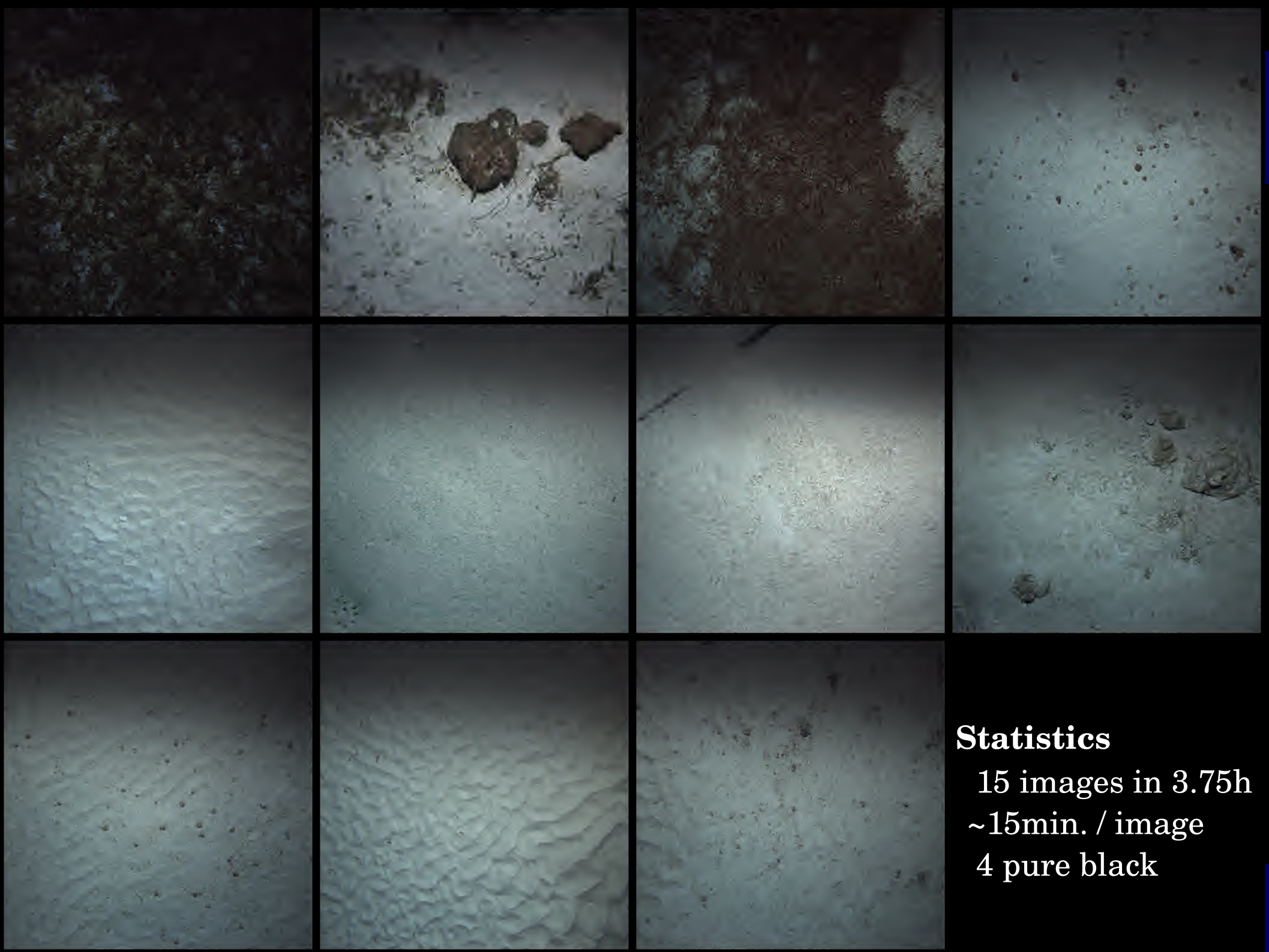
**Lack of transmission or image ID proved problematic**

# Original



# 4032 Bytes




**Statistics**
15 images in 3.75h
~15min. / image
4 pure black


1008 Bytes
2016 Bytes
3024 Bytes
4032 Bytes
5040 Bytes
6048 Bytes
7056 Bytes
8064 Bytes
9072 Bytes
10080 Bytes
11088 Bytes
12096 Bytes

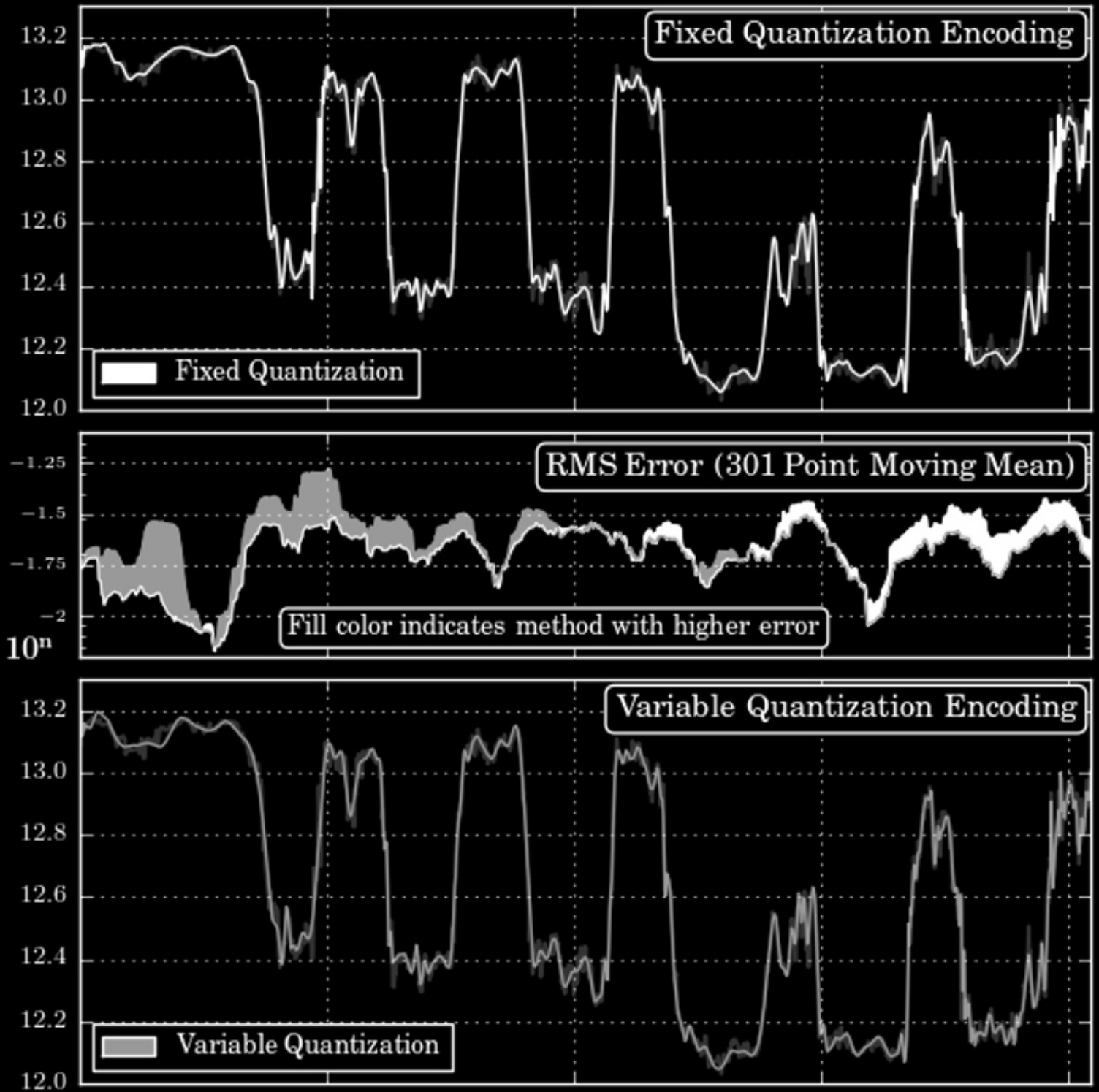
# Variable Quantization

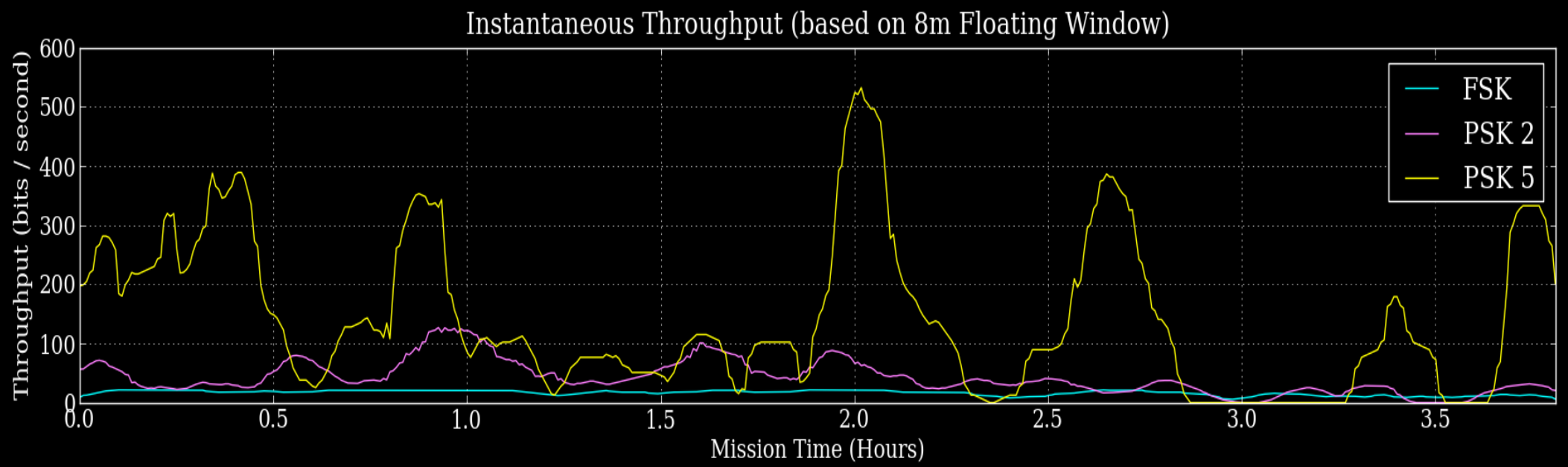
Step 1: Transform signal with Discrete Wavelet Transform.

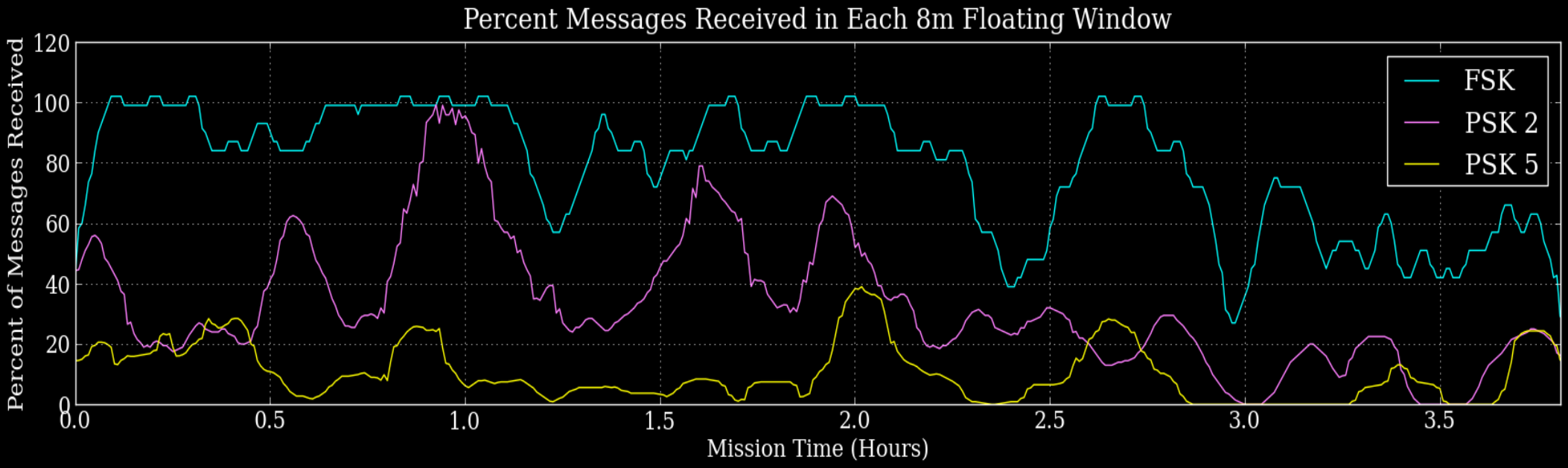
**Step 1.5: Pre-scale wavelet coefficients by objective function**

```
threshold = get_starting_threshold()
while threshold > 0:
   sort(threshold)
   refine(threshold)
   threshold >>= 1
```

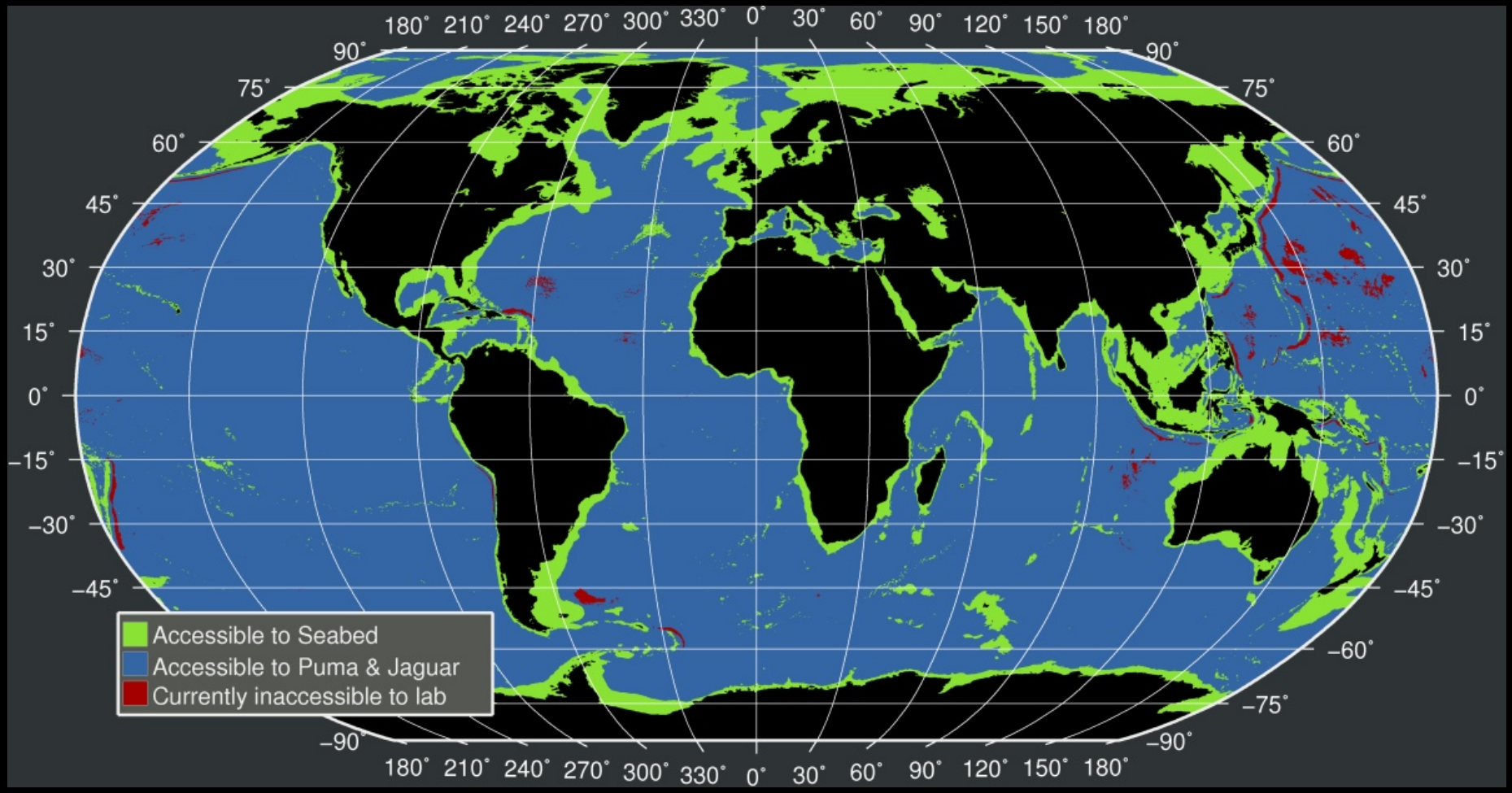
Step 2: iterates over a decreasing threshold value, identifying "significant" wavelet coefficients larger than the threshold.


Fixed Quantization Encoding
13.2
13.0
12.8
12.6
12.4
12.2
12.0
Fixed Quantization
RMS Error (301 Point Moving Mean)
−1.25
−1.5
−1.75
−2
10^n
Fill color indicates method with higher error
Variable Quantization Encoding
13.2
13.0
12.8
12.6
12.4
12.2
12.0
Variable Quantization




**Which "rate" should I use? Why must they be fixed?**




# Thank You.

chrismurf @ whoi.edu