Data Compression of Science and Housekeeping data for Planetary Probe Missions

M. Cabral, R. Trautner, R. Vitulli, C. Monteleone

TEC-ED, ESA-ESTEC
Overview

Data Compression – General
Benefits and Implementation Requirements
Available Standardized Algorithms
New Algorithms under Standardization
Data pre-Processing / Performance Optimization
Data Compression Evaluation Tools
Summary
Data Compression -

General

**Data Compression**
Data compression is the process of encoding information for reducing data volume through use of specific encoding schemes

**Lossless Compression**
Lossless compression algorithms are reversible and exploit statistical redundancy in such a way as to represent the original data more concisely without error.

**Lossy Compression**
Lossy compression discards some of the information contained in the original data in order to achieve additional data volume reductions. Lost information cannot be recovered.

Noise (random data) is incompressible with lossless algorithms. However, as most measurement data contains noise, lossy compression can often be used to achieve significant data volume reduction without losing the desired information content.
Benefits and Implementation

Requirements

**Data Compression Advantages**

Reduced data volume leads to
- Reduced onboard memory / storage requirements
- Reduced telemetry bandwidth requirements
- Reduced transmission power and indirect mass savings on related systems

**Implementation Options**

- Software (usually in particular for control processing type algorithms)
- Hardware (often for stream processing / high bandwidth applications)

**Implementation Requirements**

- Hardware: availability of dedicated hardware components (ASIC) or programmable devices (FPGA) in combination with compression IP
- Software: moderate memory, processing power required

For Exploration missions including planetary probes (comparatively low data volumes, mass constraints), software compression is generally more attractive.
Standardized Compression Algorithms: Overview

**Standardization through CCSDS**
- Selects and specifies performant, resource efficient algorithms
- Supports proposed algorithm via documentation, test data, and code

**Lossless Compression**
- CCSDS 121.0-B-1 algorithm for lossless (RICE) data compression
- Generic, lossless, low complexity

**Image Compression**
- CCSDS 122.0-B-1 algorithm for lossless & lossy image compression
- Low complexity and memory requirements, fixed / floating point

**Multispectral Data Compression**
- Lossless algorithms: standardization process almost complete
- Probably 1 ESA (+error resilient & +parallelizable) 1 JPL algorithm (+performant)
- Lossy algorithms: standardization process planned next (1-2 years)
Benchmark platform: SpW-RTC (LEON2)

SpW Remote Terminal Controller (SpW-RTC) board

- SPARC V8 LEON2-FT processor with FPU
- 50MHz core clock frequency
- 34 Dhrystone MIPS
- SpW links (RMAP compliant)
- CAN controller
- 64Kbyte on-chip SRAM (EDAC protected)
- 16 Mbyte RAM
- 16 Mbyte PROM

=> representative for today’s space processors
Benchmark dataset

Test images used for benchmarks:

- **CCSDS data test images:**
  marstest.raw, sun_spot.raw, b1.raw and solar.raw

- **Synthetic images:** black and noisy

* ESA TN TEC-EDP/2008.18/RT, “Next Generation Space DSP Software Benchmark”
Compression Test Results (1)

- **CCSDS 121.0-B-1 algorithm for** lossless generic data compression
- **Platform:** SpW RTC ASIC, 50 MHz

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>imageBlack1024x1024x12.raw-LL</td>
<td>2097152</td>
<td>100352</td>
<td>20.90</td>
<td>2.55</td>
<td>1.27</td>
</tr>
<tr>
<td>marstest512x512x8.raw-LL</td>
<td>262144</td>
<td>173397</td>
<td>1.51</td>
<td>1.84</td>
<td>7.38</td>
</tr>
<tr>
<td>solar.raw1024x1024x12-LL.wd</td>
<td>2097152</td>
<td>956401</td>
<td>2.19</td>
<td>10.92</td>
<td>5.46</td>
</tr>
<tr>
<td>imageRandom1024x1024x12.raw-LL</td>
<td>2097152</td>
<td>1747628</td>
<td>1.20</td>
<td>14.16</td>
<td>7.08</td>
</tr>
</tbody>
</table>

⇒ Noisy image is worst case for compression time
⇒ Compression time increases ~linear with # pixels
⇒ Performance typical 7-11 seconds per Mpixel / 5-8 seconds per Mbyte
Compression Test Results (2)

- **CCSDS 122.0-B-1 algorithm for lossless image compression**
- Platform: SpW RTC 50 MHz

⇒ Again, noisy image is worst case for compression time
⇒ Compression time scales with # pixels
⇒ **Pixel depth (8/12/16bit etc) not important (byte / word access time similar)**
⇒ **Performance ca 37-52 seconds/Megapixel**
Comparison of LOSSLESS CCSDS compression algorithms

- CCSDS algorithm for lossless (RICE) data compression (CCSDS 121.0-B-1)
- CCSDS algorithm for lossless image compression (CCSDS 122.0-B-1)

![Comparison - RICE vs lossless image compression](image)

<table>
<thead>
<tr>
<th>Image</th>
<th>COMP ratio</th>
<th>SPEED ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>imageBlack1024x1024x12.raw-LL</td>
<td>40.98</td>
<td>7.01</td>
</tr>
<tr>
<td>marstest512x512x8.raw-LL</td>
<td>1.11</td>
<td>6.46</td>
</tr>
<tr>
<td>solar.raw1024x1024x12-LL.wd</td>
<td>1.18</td>
<td>4.70</td>
</tr>
<tr>
<td>imageRandom1024x1024x12.raw-LL</td>
<td>1.05</td>
<td>5.06</td>
</tr>
</tbody>
</table>

=> For realistic images, **RICE is ~5-6 times faster on same platform**
=> For realistic images, **121.0-B-1 compresses ~10-20% more efficiently**

=> Tradeoff to be done for each application case
Compression Test Results (4)

- **CCSDS 122.0-B-1 algorithm for lossy image compression**
- **Platform:** SpW RTC ASIC, 50 MHz

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>imageBlack1024x1024x12.raw-LL</td>
<td>2097152</td>
<td>2449</td>
<td>856.33</td>
<td>20.25</td>
<td>10.12</td>
</tr>
<tr>
<td>marstest512x512x8.raw-LL</td>
<td>262144</td>
<td>26176</td>
<td>10.01</td>
<td>7.02</td>
<td>28.10</td>
</tr>
<tr>
<td>solar.raw1024x1024x12-LL.wd</td>
<td>2097152</td>
<td>104832</td>
<td>20.00</td>
<td>28.05</td>
<td>14.03</td>
</tr>
<tr>
<td>imageRandom1024x1024x12.raw-LL</td>
<td>2097152</td>
<td>104832</td>
<td>20.00</td>
<td>22.40</td>
<td>11.20</td>
</tr>
</tbody>
</table>

=> Noisy image NOT worst case
Data volume available is filled faster for noisy images
images with moderate entropy take longer

=> Compression time scales with # pixels, depth (8/12/16bit etc) not important

=> **Performance for realistic images** (@ 0.8 bits per pixel) **ca 27 to 30 sec / Msample**
Lossy / Near-lossless Multispectral / Hyperspectral Compression

- Candidate algorithm for multispectral data compression standard developed by ESA contractor
- Low complexity, good error resilience (error propagation restricted to 16x16 pixel columns), high compression
- Very efficient hardware implementation, good parallelizability

Prototype FPGA resource consumption

<table>
<thead>
<tr>
<th>Device</th>
<th>Xilinx xc4vlx200</th>
<th>Xilinx xc2v3000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Used LUT</td>
<td>10306 (5%)</td>
<td>10248 (35%)</td>
</tr>
<tr>
<td>Ram 16s</td>
<td>21 of 336 (6%)</td>
<td>21 of 96 (22%)</td>
</tr>
<tr>
<td>Multi18x18s</td>
<td>9 of 96 (9%)</td>
<td>9 of 96 (9%)</td>
</tr>
<tr>
<td>DSP48</td>
<td>81</td>
<td>79</td>
</tr>
<tr>
<td>Max freq (MHz)</td>
<td>70</td>
<td>69</td>
</tr>
<tr>
<td>Throughput (M samples/sec)</td>
<td>69</td>
<td>69</td>
</tr>
</tbody>
</table>

Standardization process ongoing

<table>
<thead>
<tr>
<th></th>
<th>Proposed (Golomb)</th>
<th>Proposed (GPO2)</th>
<th>LUT</th>
<th>FL</th>
</tr>
</thead>
<tbody>
<tr>
<td>sc0</td>
<td>6.44</td>
<td>6.45</td>
<td>7.14</td>
<td>6.23</td>
</tr>
<tr>
<td>sc3</td>
<td>6.29</td>
<td>6.30</td>
<td>6.91</td>
<td>6.10</td>
</tr>
<tr>
<td>sc10</td>
<td>5.61</td>
<td>5.62</td>
<td>6.26</td>
<td>5.65</td>
</tr>
<tr>
<td>sc11</td>
<td>6.02</td>
<td>6.04</td>
<td>6.69</td>
<td>5.86</td>
</tr>
<tr>
<td>sc18</td>
<td>6.38</td>
<td>6.39</td>
<td>7.20</td>
<td>6.32</td>
</tr>
<tr>
<td>Average</td>
<td>6.15</td>
<td>6.16</td>
<td>6.84</td>
<td>6.03</td>
</tr>
</tbody>
</table>

Compression performance (bits/pixel) in comparison to other algorithms (LUT and FL)
Improvement of lossless RICE compression by data transposition

- **Data transposition for constant structure & size packets:** preparation step for high efficiency RICE compression (transposition possible on bit-, byte-, and word level; byte / word level preferable)

**Constant size / structure packet (typical for HK data):**

**Data file of N packets:**

![Data file of N packets]

**Data file of 8 packets, transposed:**

![Data file of 8 packets, transposed]

⇒ Data compression with RICE algorithm (lossless) much more efficient!
⇒ Typical improvement: from factor 1... 2 to factor 5 ... > 10

Example: VEX magnetometer HK data (700kbytes = ca 7000 packets):
Compress improves from 1.09 => 13.27
Impact of \# of packets on compression performance

Example:
• VEX magnetometer HK data, 5 \ldots 5000 packets
• Packet size 98 bytes, 16-bit transposition, words 16 bit unsigned
• RICE compression, settings: block size 8 / 16 bit, ref insertion 64

⇒ Even for moderate packet numbers (20 \ldots 200) a lossless compression ratio of \approx 4 \ldots 10 is achievable

⇒ Transposition is fast: 0.27 sec for a 1 Mword TM packet sequence on a 50 MHz RTC (LEON2)
WhiteDwarf data compression evaluation tool

- Allows potential users to test the recommended algorithms
- Software is made available via ESA TEC-EDP
- A choice of standardized algorithms is available, more to be added in future (multispectral image compression)
- Supports both compression + decompression for easy verification
- Several data pre-processing methods are also implemented
- Application is both WIN and Linux based, executable, for distribution to payload teams / industry: to be used for tests with user data
- Teams / industry can play with processing and compression algorithms and their parameters, select pre-processing steps, compression algorithm and best settings for their type of data
Tools for Users – *WhiteDwarf* (2)

**WhiteDwarf GUIs**

[Image of WhiteDwarf GUIs]

14-18 June 2010
Summary (1)

**Compression can be enabling technology for missions**
- Data volume reduction, resource saving, bandwidth reduction

**Standardized compression algorithms available**
- Resource efficient, low complexity, good performance, well supported

**Benchmarks performed on state of the art processor**
- Data and application characteristics important
- Benchmark results are available

**Pre-processing significantly improves packet data compression**
- Highly recommended for HK / SCI data on probe missions
- Can easily be combined with standardized lossless algorithms
- Very high lossless compression achievable

**Compression algorithm evaluation tool is provided**
- Made available by TEC-EDP, ESA-ESTEC
For a copy of Whitedwarf –
- Contact Roland.Trautner@esa.int, or
- Contact Raffaele.Vitulli@esa.int

For data compression & transposition
- Have a look at our IPPW7 paper
- Also look at
  SpaceOps Conference, 2010