Efficient Image Coding and Transmission In Deep Space Communication

Reiner Dizon
Department of Electrical and Computer Engineering, University of Nevada, Las Vegas

Introduction

In deep space communication framework, compression for reducing the bandwidth/time for transmitting data to the ground is to be performed losslessly. Three compression algorithms known for their efficiency of compression, hardware simplicity and error resilience (recovery from bit changing) are Huffman, Rice (on CCSDS 121.0-B-1 standard) and Exp-Golomb are considered for the quantitative comparison. For on-board (satellite) implementation other parameters of paramount importance are component use and power.

Goal of this study: implement and evaluate Huffman, Rice and Exp-Golomb methods for on-board image compression.

Methods

- Test images include benchmark and astronomical images with 8-bit and 16-bit data resolution.
- Hardware implementation of each algorithm done in Verilog and programmed to FPGA.
- Huffman circuit includes histogram calculation, sorting, codeword length calculation, and canonical tree construction.
- Rice circuit based on CCSDS 121.0-B-1 standard with 8-bit and 16-bit data along with block size of 8 and 16.
- Repetition code added to the header of Rice code.
- Exp-Golomb circuit uses Rice's preprocessor and LUTs.
- Software implementation done in C++ & MATLAB to assess decoded data with or without errors.
- Errors introduced using random number generator.
- Quartus II provides data for the components usage and power consumption.

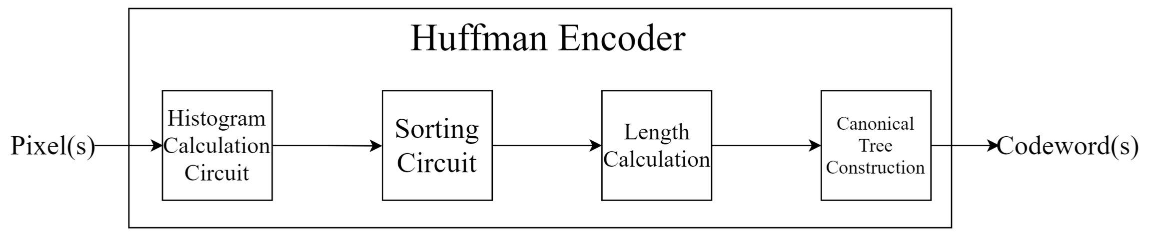


Fig. 1 – High-Level Schematic of Huffman Circuit

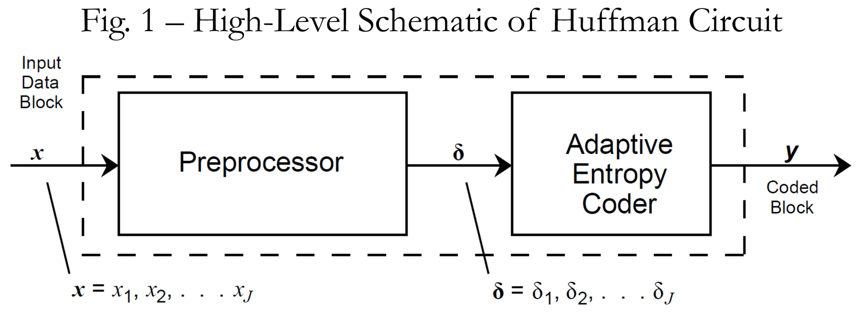
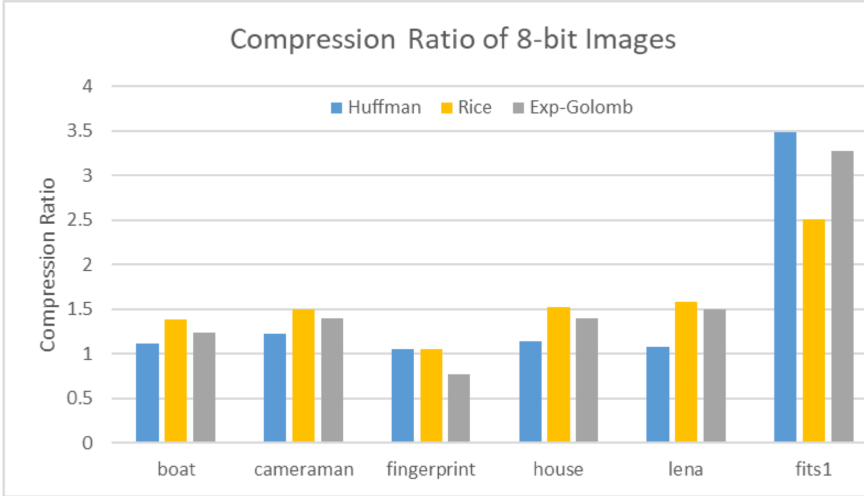
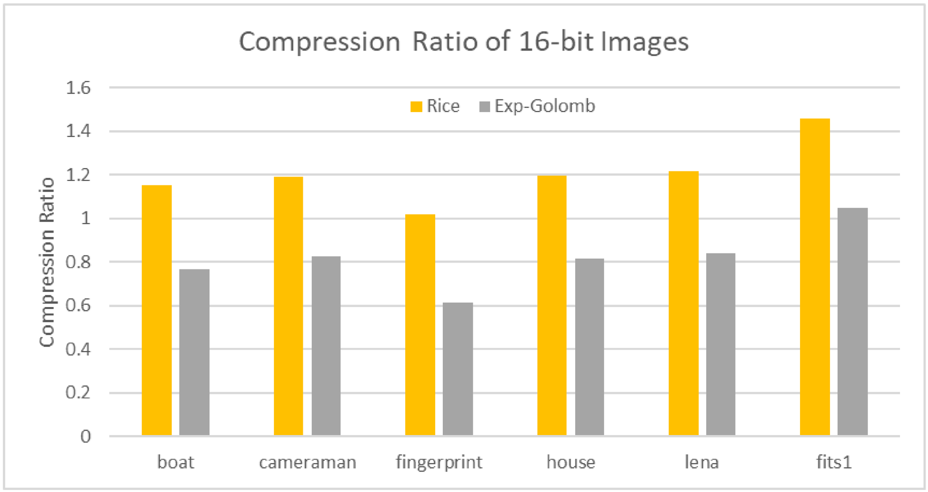


Fig. 2 – Main Functional Parts of Rice Encoder (Source: CCSDS, 2012)

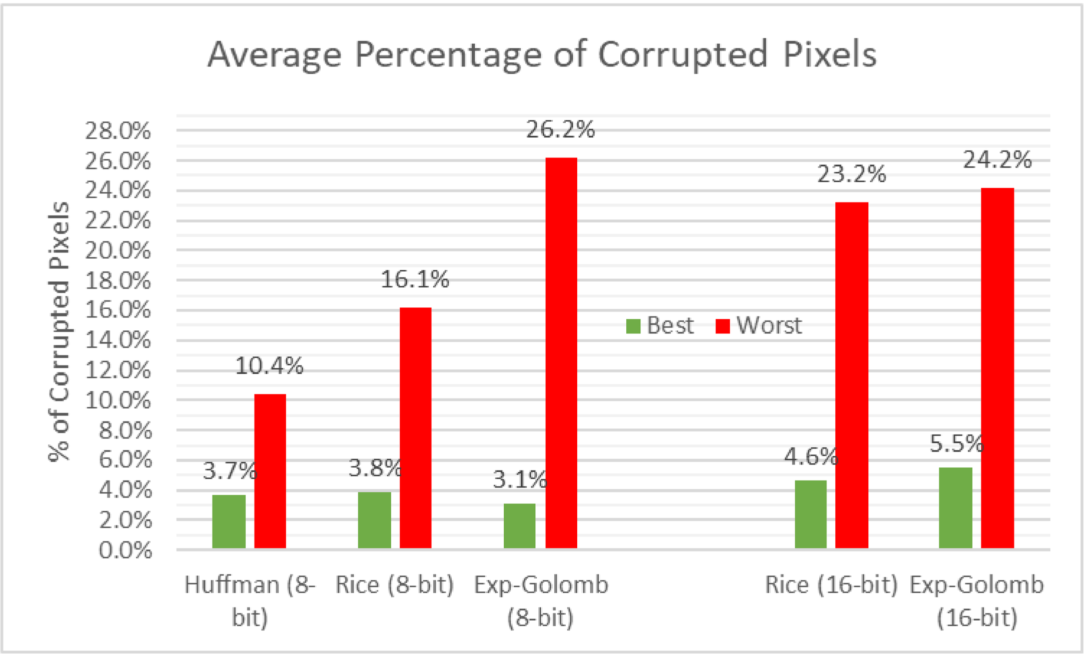
Analysis

Compression Ratio = Number of bits in the original image/Number of bits in the compressed image

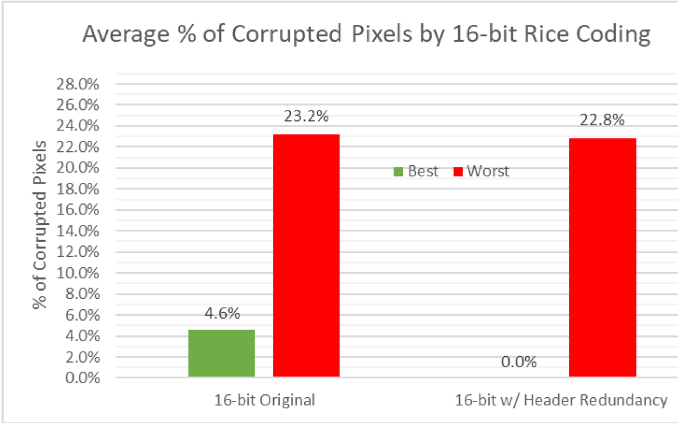




Error Resilience is assessed by the number of corrupted pixels in the decoded image.



Need to protect the Rice block header from errors



Component usage and power

Components	total logic elements	total combinational functions	total registers
Huffman (4-bit)*	308	300	177
Rice (n = 8, J = 8)	338	330	56
(header_red)	336	335	56
Rice (n = 8, J = 16)	426	426	58
(header_red)	434	434	58
Rice (n = 16, J = 8)	517	517	74
(header_red)	557	541	74
Rice (n = 16, J = 16)	1130	1130	75
(header_red)	1152	1136	75
Exp-Golomb (8-bit)	354	346	72
Exp-Golomb (16-bit)	759	758	89

Circuit	Estimated Power Dissipation (mW)
Huffman (4-bit)*	215.25
Rice (n = 8, J = 8)	172.77
(header_red)	176.87
Rice (n = 8, J = 16)	181.20
(header_red)	177.28
Rice (n = 16, J = 8)	193.31
(header_red)	189.76
Rice (n = 16, J = 16)	214.94
(header_red)	214.98
Exp-Golomb (8-bit)	175.56
Exp-Golomb (16-bit)	190.11

Conclusion

- Rice coding exhibits the best performance in terms of compression ratio for both 8-bit and 16-bit images.
- Huffman and Rice are both error resilient for 8-bit and 16-bit astronomical image, respectively.
- Header redundancy on Rice coding slightly
- improved error resilience.
- For block size of 8, Rice encoder uses fewer components than the Exp-Golomb method does.
- The simplicity of Exp-Golomb implementation results in the least power consumption among the circuits.
- Huffman coding for the given alphabet size is impractical for hardware implementation.
- Rice encoder circuit with block size of 8 provides the best solution.
- Overall, CCSDS 121.0-B-1 augmented by the repetetaion code of the header and block size of 8 offers the best balance among the circuit configurations.

Acknowledgments

- My deepest gratitude to my advisor, Dr. Emma Regentova, and committee members, Dr. Andrew Hanson and Dr. Venkatesan Muthukumar.
- This work is partially supported by NSHE/NASA grant #18-28.