



Nunez-Yanez, J.L., Chen, X., Canagarajah, C.N., & Vitulli, R. (2008). Statistical lossless compression of space imagery and general data in a reconfigurable architecture. In *NASA/ESA Conference on Adaptive Hardware and Systems (AHS 2008)*, Noordwijk, Netherlands (pp. 172 - 177). Institute of Electrical and Electronics Engineers (IEEE).
<https://doi.org/10.1109/AHS.2008.9>

Peer reviewed version

Link to published version (if available):
[10.1109/AHS.2008.9](https://doi.org/10.1109/AHS.2008.9)

[Link to publication record on the Bristol Research Portal](#)
PDF-document

University of Bristol – Bristol Research Portal

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available:
<http://www.bristol.ac.uk/red/research-policy/pure/user-guides/brp-terms/>

Statistical Lossless Compression of Space Imagery and General Data in a Reconfigurable Architecture

Jose Luis Nunez-Yanez, Xiaolin Chen, Nishan Canagarajah

*Electronic Engineering Department
Bristol University
Bristol, BS8 1UB, UK
{eejlny,eezxxc,eecnc}@bristol.ac.uk*

Raffaele Vitulli

*European Space Agency (ESA)
On-Board Payload Data Processing Section
Keplerlaan 1, Noordwijk, 2200, The Netherlands
raffaele.vitulli@esa.int*

Abstract

This paper investigates an universal algorithm and hardware architecture for context-based statistical lossless compression of multiple types of data using FPGA (Field Programmable Gate Array) devices which support partial and dynamic reconfiguration. The proposed system enables optimal modeling strategies for each source type whilst entropy coding of the modeling output is performed using a statically configured arithmetic coding engine. Spacecraft communications typically involve large amounts of information captured from different sensors that must be transmitted without any loss. The statistical redundancies present in this data can be removed efficiently using the proposed reconfigurable compression technology.

1. Introduction

Lossless and lossy compression algorithms are routinely used to reduce the bandwidth and storage requirements of digital data. Lossy compression is well suited to data that is already a digital approximation of data that is analogue in nature such as visual and audio information. Lossy compression can achieve much higher compression ratios than lossless precisely because there is not a requirement to maintain all the information contained in the data source. Lossy compression has been widely adopted and global standards exist such as H.264 and MPEG4 for video and JPEG for still images. Traditionally, lossless compression has been used in those data types that do not admit any bits to be modified in the compression/decompression processes. Examples are general data types such as text, html code, database information, application data and program binaries where reversible compression is required since every bit contains critical information. Nevertheless, lossless compression of images and video is also an important topic of research. Medical images such as X-rays must be compressed without any loss since the implications could be catastrophic for the patient. Precious and hard-to-acquire images such as those obtained in space exploration and satellite surveillance also use lossless compression. Image archiving will benefit from a master copy stored using a

lossless approach from which copies of any desired quality could be obtained using lossy methods. Additionally applications such as data, video and image transmission in space require the performance to be achieved in an energy and silicon efficient platform. To achieve the demands set by these applications we propose a universal lossless compression hardware core that exploits dynamic reconfiguration to effectively combine predictive coding, motion estimation and context-based modeling hardware blocks depending on the data type.

2. Related work

Current lossless compression for general data makes a distinction between dictionary-based and statistics-based algorithms. Dictionary-based compression has been traditionally more popular in software and hardware due to the inherent simplicity of these algorithms. Examples of dictionary-based compression in software are the popular WinZip or Arj algorithms commonly used for archiving and distributing large quantities of data. Also, the hardware devices available from leading companies such as HiFn Microelectronics LZS [1] use dictionary-based compression methods based on the original LZ [2] algorithms. Our research group's work has been extensively based on dictionary-based compression with the X-MatchPRO [3] algorithm targeting a compression ratio that halves the original input size while operating at very high speeds. This research area both in software and hardware is now mature with more than 25 years of work having being dedicated to fine tune the LZ method and few improvements can be expected. Context-based statistical compression has been limited to software and it is not very popular due to its high computing requirements. An example of a powerful statistical compressor is the PPMZ [4] software algorithm that requires more than 20,000 CPU cycles per byte in a general purpose microprocessor.

Algorithmic research in lossless image compression has focussed in two main techniques; transform-based and predictive coding. Extensive experimentation seems to indicate that transform methods perform worse than predictive methods for lossless compression. Predictive

coding is a technique where the value of the next input data is predicted as a linear or nonlinear combination of previous inputs. It has been successfully used in lossless compression of visual content where it makes use of a priori knowledge of smoothness. Smoothness means that visual signals tend to follow a pattern of gentle variation that can successfully be exploited in the first processing stage to reduce the entropy of the data source. The output of the predictive coder can be modelled using a context dependent technique to further remove redundancy prior to being entropy coded using Huffman coding or arithmetic coding. Algorithms that employ this approach are the Sunset [5], Felics [5] and the LOCO [6] algorithm used in the JPEG LS. There are few hardware devices proposed for lossless image compression using predictive context-based arithmetic coding. A successful example targeted at the compression of black and white fax images is the IBM Q-Coder [7] device. This chip is based on a simple fixed high-order (7th) binary model. This simple model means that performance is not well suited for general alphabets. This device achieves a throughput of 64 Mbits/second when implemented in a CMOS 5S (0.35 μm) technology from IBM.

Hardware-based lossless video compression is a largely unexplored area. An example of a software lossless video codec is the Huffvuv algorithm heavily based on lossless JPEG operating on each of the frames of the sequence. Popular lossy video codecs such as H.264 also offer a lossless mode [8].

3. Dynamically Reconfigurable Modeling Stage

The proposed compression system uses a dynamically reconfigurable modeling stage followed by statically configured probability estimation and arithmetic coding stages as illustrated in Fig.1. Dynamic modeling is specialized to each data type and uses a combination of context modeling, predictive coding and motion estimation depending on the data type being processed: 1-D general data, 2-D image data or 3-D multispectral images or video.

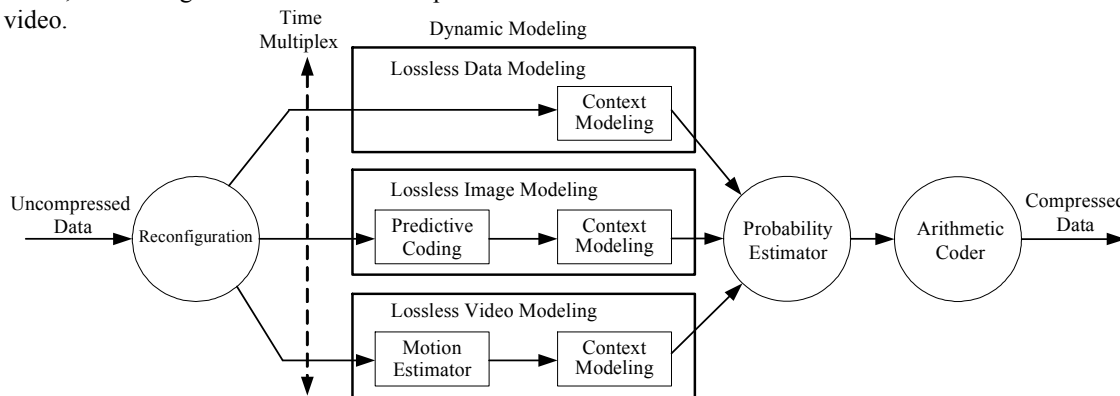


Figure 1. System Overview

3.1. 1-D Lossless Data Modeling overview

During context modeling for 1-D data a finite number of symbols (model order) that preceded the current symbol in a single dimension and constitute its context are searched in a context tree built dynamically as more data is seen. Fig. 2 shows a simplified diagram of the context modeller. The context FIFO stores the symbols that preceded the current symbol and form its context. The FIFO width is 1 byte to match the width of the symbol while its length is configurable and depends on the maximum model order.

The hardware implementation of the context modeler is based on a hashing tree that enables fast search operations with low complexity. The tree is stored in standard SRAM memory and maintains its logical structure using a pointer mechanism. The hashing shift and the XOR gate in Fig. 2 are used to generate an index to be used to address the SRAM memory that stores the context tree.

The tree memory is divided into three sections. Section 1 stores the context area memory address where the probability data for that particular tree node can be found. The other two sections implement the pointer mechanism that maintains the logical structure of the tree. Section 2 stores the context area index of the tree node parent of the current node in the tree structure. Section 3 stores the context symbol stored at the current tree node. The SRAM area free memory and the busy area generator shown in Fig. 2 enable a single-cycle reset state without having to reset the table memory with a multi-cycle table walk operation. A table walk would have had a very negative effect on throughput when dealing with small blocks since the number of cycles needed to reset the table could typically be larger than the number of cycles needed to compress the block. A single valid register, named line free in Fig. 2, is reset after processing each block and this automatically invalidates all the locations in the table memory.

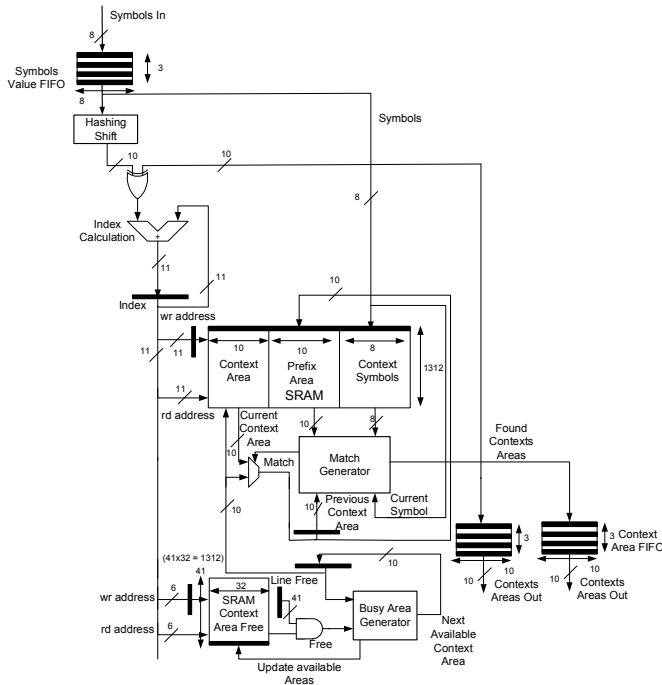


Figure 2.1-D general data modeling architecture

This register has a similar function to the register holding the valid bits in a direct-mapped cache. Each of the register bits is shared by several table locations and in order to distinguish which context tree nodes are busy and which context tree nodes are free the area free memory contains 1 bit per context tree node signaling a free or busy node. If the valid register bit is set to zero all the tree nodes associated with that valid register bit are considered invalid. The found context areas are stored in two equivalent buffers. When the first buffer is being filled with context areas by the context modeler the second buffer is being emptied by the probability estimator. Once both stages have completed their operation the buffers functionality is reversed and the process restarted. This double buffering mechanism increases the throughput of the system avoiding idle stages.

3.2. 2-D Lossless Image Modeling overview

Lossless image modeling handles image or any data which has two-dimensional correlations. We propose a segmentation-based lossless image model. Segmentation, here means partitioning of an image into multiple regions according to its features. We use this idea to group pixels with similar features and use different modes to compress them. A new ternary-mode is proposed to detect and encode the edges, while the run-length coding [6] is adopted to encode the homogeneous regions. The rest of the image, mostly the texture regions, is compressed with a regular-mode, which is based on the Gradient-Adjusted

Prediction (GAP) from CALIC [9] but is simplified. As the mode selection is made by adaptive online checking of neighboring symbols, no side information is transmitted.

We identify certain conditions for entering each mode. If the four nearest symbols of the current symbol are the same, a homogeneous region is assumed and the run-mode is triggered. If the current symbol is identical to its previous symbol, the symbol occurrence, called run, increases by one; otherwise it stops and the current run length is encoded. In regions where edges are present, we examine if there are no more than three distinct symbol values in a small neighborhood of the current symbol and the ternary-mode is triggered. Thus only four symbols are needed to encode this group of symbols and lower entropy can be obtained. When the entry conditions for run-mode or ternary-mode cannot be met, or when coding in other modes fails, the regular mode is used.

Fig.3 illustrates the dataflow of the image model. The implementation is achieved with two pipelines running in parallel. Line 1, indicated by the flow on the left, operates on the current symbol and yields the prediction error with the selected mode for the probability estimator.

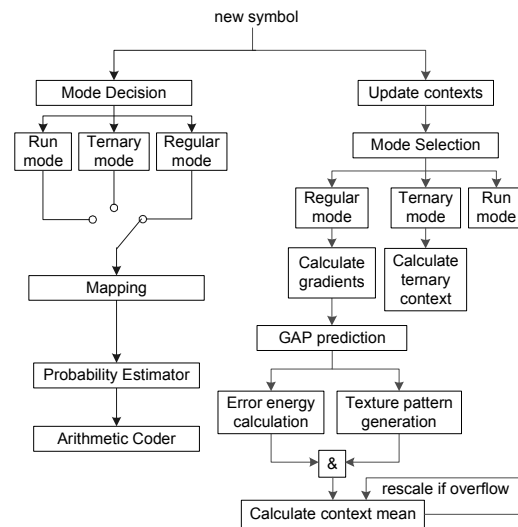


Figure 3. 2-D data modeling architecture

Line 2, indicated by the flow on the right, calculates the prediction value and context index for the next symbol under the selected mode. Since complicated coefficient calculations are not needed, and simple division is done by small lookup table, this model is hardware amenable. This model is the base of the video model and can be extended to handle multispectral images. Based on the 2-D model, the video model incorporates the decorrelation in spectral domain and temporal domain. An inter-band prediction is used to exploit the correlation in spectral domain and a switching strategy is designed to switch

while increasing its suitability for hardware implementation. The resulting MZ-coder balances the complexity of coding the MPS and LPS symbols, simplifies the precision of the arithmetic and handles special hardware borrow conditions while maintaining coding efficiency and achieving high performance.

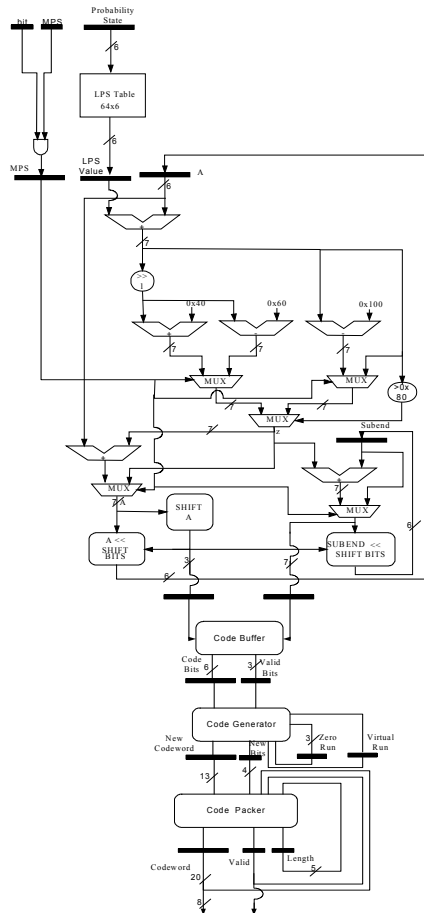


Figure 5. Arithmetic coding architecture

Fig.5 shows the internal organization of the multiplication-free arithmetic coding module. A total of 6 pipeline stages are identified to improve the clock ratio of the design. The lack of a renormalization loop in the MZ algorithm means that one decision bit is processed per clock cycle. The functionality of each of the pipeline stages is explained in detailed in [10].

5. Performance Comparison

This section analyses the performance of the core in terms of compression ratio and throughput and compares it with other compression algorithms for data and images implemented in both hardware and software.

Table 1 compares the compression efficiency of the algorithm configured in data mode with two dictionary-

based and two statistical-based algorithms using the Canterbury corpus as the data set. We have selected the popular open source Lempel-Ziv implementation known as GZIP and equivalent to other commercial algorithms such as PKZIP and WinZIP as a fast and efficient dictionary-based algorithm. LZS is targeted to hardware as described in the related work section.

File	GZIP	LZS	PPMC	PPMZ	Proposed (data mode)
Alic	2.86	4.19	2.82	2.08	2.43
Asyo	3.05	4.29	3.01	2.26	2.57
Cppp	2.54	3.64	2.50	2.14	2.48
Fiel	2.20	2.87	2.14	1.81	2.16
Gram	2.62	3.09	2.57	2.25	2.51
Kenn	1.60	2.31	1.42	1.08	1.37
Lcet	2.71	4.14	2.69	1.82	2.25
Plra	3.24	4.69	3.22	2.21	2.49
ptt5	0.87	1.26	0.82	0.79	0.89
Sum	2.70	3.54	2.59	2.46	3.10
Xarg	3.29	3.89	3.22	2.84	3.14
average	2.52	3.45	2.45	1.98	2.30

Table 1. General data compression performance

PPMC and PPMZ are software-only complex statistical algorithms which need around 2,000 and 20,000 clock cycles on average to compress a byte when implemented in a general purpose processor. The table measures compression in bits per byte and shows that only PPMZ outperforms the proposed method. To evaluate image compression we use the 8-bit CCSDS reference image set as test images. As the proposed system is intended for space borne applications test results relevant to this purpose are useful. We compare the proposed scheme with some state-of-the-art low complexity schemes. CCSDS is the current Recommendation for space image compression; PRDC is the CCSDS Rice coder; JPEG-LS is the lossless image compression standard; JPEG2000 [11] is the current standard for lossy to lossless compression; SPIHT [12] is a low-complexity progressive image compressor; ICER [13] is another progressive wavelet-based image compressor. When strip-based and frame-based options are available for these algorithms, the best ones are chosen in the comparison. Table 2 shows that the proposed system outperforms the others in terms of bit rates. In terms of throughput performance, the proposed system is designed to process 1 bit per clock cycle, which translates into a throughput of 100Mbits/sec on a Xilinx Virtex-4 SX35 FPGA.

image	CCSDS	PRDC	JPEG-LS	JPEG2000	SPIH T	ICER	Proposed (image mode)
coastal – b1	3.36	3.56	3.09	3.13	3.09	3.07	3.00
coastal – b2	3.22	3.32	2.90	2.97	2.94	2.92	2.84
coastal – b3	3.48	3.68	3.22	3.23	3.21	3.20	3.14
coastal – b4	2.81	2.91	2.41	2.53	2.57	2.55	2.37
coastal – b5	3.16	3.30	2.81	2.94	2.91	2.89	2.79
coastal – b6h	3.02	2.75	2.50	2.60	2.71	2.54	2.52
coastal – b6l	2.35	2.03	1.76	1.96	2.02	1.87	1.84
coastal – b7	3.45	3.66	3.17	3.22	3.17	3.15	3.10
coastal – b8	3.66	3.93	3.42	3.40	3.35	3.31	3.28
europa3	6.61	7.48	6.64	6.52	6.46	6.30	6.42
marstest	4.78	5.39	4.69	4.74	4.64	4.63	4.60
lunar	4.58	5.23	4.35	4.49	4.43	4.40	4.20
spot-la – b3	4.80	5.20	4.53	4.69	4.70	4.56	4.43
spot_la – panchr	4.27	4.87	4.00	4.13	4.11	4.03	3.90
average	3.82	4.09	3.54	3.61	3.59	3.53	3.46

Table 2. Space Imagery data compression performance

6. Conclusions

The compression ratio evaluation of the algorithm shows that the proposed method can outperform other well-known techniques. The hardware amenability and the reconfigurable feature mean that the device could operate in a resource and energy constraint environment such as a space probe. In principle, reconfiguration should be initiated by a general controller although automatic reconfiguration after data type detection is also possible. We are currently working in adding lossless video compression support designing an efficient pixel-oriented vector-less motion estimation engine. Additionally, we would like to investigate the configuration of different alphabet sizes extending the current byte-based alphabets to multiple-bit alphabets for lossless compression of scientific data obtained from high-resolution analogue-to-digital converters. Executables and information for this core named Byacom-2 will become available at www.byacom.co.uk as the project progresses. We would like to acknowledge the support of EPSRC under grant number EP/D011639/1 for making this research possible.

- 1 '9600 Data Compression Processor', Data Sheet, Hi/fn Inc, 750 University Avenue, Los Gatos, CA, 1999.
- 2 J.Ziv, A.Lempel, 'A Universal Algorithm for Sequential Data Compression' IEEE Trans. Inf. Theory, Vol. IT-23, pp. 337-343, 1977.
- 3 J.L. Núñez, S. Jones, 'Gbit/Second Lossless Data Compression Hardware', IEEE Transactions in VLSI Systems (TVLSI), Vol. 11, No. 3, pp. 499-510, June, 2003
- 4 C. Bloom, 'Solving the Problems of Context Modelling', <http://www.cbloom.com/papers/index.html>, 1998.
- 5 X. Wu, 'An algorithmic study in lossless image

compression', Proceedings of the 1996 Data Compression Conference, Snowbird, Utah, pp. 150-159, April 1996.

- 6 M. Weinberger, G. Seroussi, and G. Shapiro, 'The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS', IEEE Trans. on Image Proc., vol. 9, pp. 1309-1324, Aug. 2000.
- 7 M. J. Slattery, J. L. Mitchell, 'The Qx-Coder', IBM Journal of Research and Development, Vol. 42, No. 6, pp. 767-784, 1998.
- 8 Sullivan, T. Wiegand, "Video Compression - From Concepts to the H.264/AVC Standard", Proc. the IEEE, Special Issue on Advances in Video Coding and Delivery, Vol. 93, No. 1, pp. 18-31, January 2005.
- 9 X. Wu and N. Memon, "Context-based, adaptive, lossless image coding," IEEE Trans. Commun., vol. 45, no. 4, pp. 437-444, Apr. 1997.
- 10 J. L. Nunez-Yanez and V. A. Chouliaras, "A configurable statistical lossless compression core based on variable order Markov modelling and arithmetic coding," IEEE Trans. Comput., vol. 54, no. 11, pp. 1345-1359, Nov. 2005.
- 11 Taubman, D. S., Marcellin, M. W.: JPEG2000 Image Compression Fundamentals, Standards and Practice. Kluwer. 2002
- 12 Said, A., Pearlman, W. A.: A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees. IEEE Trans. Circuits Syst. Video Technol. Vol. 6, pp. 243-250, 1996.
- 13 Kiely, A., Klimesh, M.: The ICER Progressive Wavelet Image Compressor. IPN Progress Report 42-155. pp. 1-46, 2003.