A&A manuscript no.
(will be inserted by hand later)

Your thesaurus codes are:
03.13.2; 03.20.1; 04.01.1;

Lossless Compression of Astronomical Images

M. Louys[1 2], J.L. Starck[3], and F. Murtagh[1 5]

[1] Observatoire de Strasbourg, Université Louis Pasteur, F-67000 Strasbourg, France

[2] Laboratoire des Sciences de l'Informatique, de l'Image et de la Télédétection, ENSPS, Université Louis Pasteur, F-67000 Strasbourg, France

[3] DAPNIA/SEI-SAP, CEA-Saclay, F-91191 Gif-sur-Yvette Cedex, France

[5] School of Computer Science, The Queen's University of Belfast, Belfast BT7 1NN, Northern Ireland

May 21, 1999

Abstract. In Louys et al. (1999), we compared a range of powerful compression methods – fractal, wavelets, pyramidal median, JPEG – with compression tools dedicated to astronomy such as HCOMPRESS, FITSPRESS and Mathematical Morphology, and applied these to astronomical images. Quality was quantified from visual appearance, and from photometric and astrometric measurements. Computational requirements of each method were discussed. We also reviewed the implications of Web-based storage and transmission, stressing the importance of progressive vision. In this article, we present initial results on image compression with no loss of information. We also point to the secondary, albeit very considerable, benefits for astronomical image processing in a distributed computing environment such as the Web.

1. Introduction

Compression methods dealt with in Louys et al. (1999) included the following:

1. JPEG: a widely-used single image lossy or lossless compressor, based on blockwise decorrelation of the image data, followed by use of the discrete cosine transform.
2. Wavelet transform: for compression it is clear that a pyramidal transform is necessary. Our experiments were based on a bi-orthogonal Antonini-Daubechies transform.
3. Fractal coding: the image is decomposed into blocks, and each block is fractal coded.

4. Mathematical morphology: shape information is separated out in such a way the reconstruction of the input image data is possible. This is followed by the usual quadtree and Huffman coding.
5. HCOMPRESS: based on the Haar wavelet transform, this was developed at Space Telescope Science Institute by R. White, following earlier work by G. Richter.
6. HCOMPRESS and iterative refinement: following work by A. Bijaoui, to cater for block artifacts produced by HCOMPRESS, the output decompressed image can be iteratively improved. This allows the very efficient Haar wavelet transform to be kept in the compressor.
7. Pyramidal median transform: a multiscale (pyramidal) median transform is the basis of this method (Starck et al., 1996, 1998). The median provides for robustness. While there is a small price to be paid in the compressor's time requirements, we avoid any serious obstacle by making good use of the decrease of each resolution scale's dimensionality in the pyramid.

Table 1 summarizes our findings. It includes whether there is artifact creation (e.g. blockiness), and indicates availability of implementations for progressive transmission.

2. Lossless Compression

The compression methods looked at above involve filtering of information which is not considered to be of use. This includes what can demonstrably (or graphical) be shown to be noise. Noise is the part of information which is non-compressible, so that the residual signal is usually very highly compressible. It may be feasible to store such noise which has been removed from images on backing data store, but clearly the access to such information is less than straightforward.

If instead we seek a compression method which is guaranteed not to destroy information, what compression ratios can be expected?

We note firstly that quantization of floating (real) values necessarily involves some loss, which is avoided if we work in integer arithmetic only. This is not a restrictive assumption. Appropriate rescaling of image values may be availed of.

Next we note that Sweldens lifting scheme (Sweldens, 1996) provides a convenient algorithmic framework for many wavelet transforms. The low-pass and band-pass operations are replaced by predictor and update operators at each resolution level, in the construction of the wavelet transform. When the input data consist of integer values, the wavelet transform no longer consists of integer values, and so we redefine the wavelet transform algorithm to face this problem. The predictor and update operators use, where necessary, a floor truncation function. The lifting scheme formulas for prediction and updating allow this to be carried out with no loss of information. If this had been done in the usual algorithmic framework, truncation would entail some small loss of information.

For comparison purposes, we use JPEG in lossless mode. We also use the standard Unix gzip command, which implements the Lempel-Ziv run-length encoding scheme, especially widely-used for text data.

As in Louys et al. (1999), we base our experiments on image data from a Schmidt photographic plate in the region of M5 (numbered ESO 7992v), scanned with the CAI-MAMA facility. We use subimages of dimensions 1024×1024. Sampling is 0.666 arcseconds per pixel. Ancillary catalog information was available for our tests. The images are representative of those used in Aladin, the reference image service of the CDS (Strasbourg Data Centre, Strasbourg Observatory).

Table 2 shows our findings, where we see that our integer and lifting scheme wavelet transform approach is a little better than the other methods in terms of compression ratio. Not surprisingly, the compression rate is not large for such lossless compression. Routine `mr_lcomp` (MR/1, 1999) was used for this work. Lossless JPEG suffers additionally from rounding errors.

We now turn attention to usage of a lossless wavelet-based compressor, above and beyond the issues of economy of storage space and of transfer time. The lifting scheme implementation of the Haar wavelet transform presents the particularly appealing property that lower resolution versions of an image are *exactly* two-fold rebinned versions of the next higher resolution level. For aperture photometry and other tasks, lower level resolution can be used to provide a partial analysis. A low resolution level image can be used scientifically since its "big" pixels contain the integrated average of flux covered by them.

The availability of efficiently delivered low resolution images can thus be used for certain scientific objectives. This opens up the possibility for just one innovative way to analyze distributed image holdings.

3. Conclusion

We have presented results for lossless astronomical image compression, including a new integer compressor based on Sweldens lifting scheme. We have benchmarked this and other compression methods. We have also indicated one novel analysis strategy permitted by our compression approach, which is feasible in a distributed computing environment.

Acknowledgements

G. Skalkowski carried out experiments on the lossless compression in the context of a DEA *stage* at Strasbourg Astronomical Observatory.

References

Louys M, Starck JL, Mei S, Bonnarel F & Murtagh F, 1999, Astronomical image compression, A&AS, 136, 579–590.

MR/1: The Multiresolution Analysis Software, Version 2, 1999, http://visitweb.com/multires

Sweldens W, 1996, The lifting scheme: a custom-design construction of biorthogonal wavelets, Appl. Comput. Harmon. Anal., 3, 186-200.

Starck JL, Murtagh F, Pirenne B & Albrecht A, 1996, Astronomical image compression based on noise suppression, PASP 108, 446–455.

Starck JL, Murtagh F & Bijaoui A, 1998, Image and Data Analysis: The Multiscale Approach, Cambridge University Press.

	Comp. time (sec)	Decomp. time (sec)	Artifact	Comp. ratio	Progressive transmission
JPEG	1.17	4.7	Y	<40	Y (in C)
Wavelet	45	7.1	Y	270	N
Fractal	18.3	9	Y	< 30	N
Math. Morpho.	13	7.86	N	< 210	N
Hcompress	3.29	2.82	Y	270	Y (in C)
Hcompress + iter rec	3.29	77	N	270	N
PMT	7.8	3.1	N	270	Y (in Java)

Table 1. Compression of an 1024 $\times$ 1024 integer-2 image. Platform: Sun Ultra-Enterprise, 250 MHz and 1 processor.

Software	CPU time (sec) Compression time	Decompression time	Compression ratio
JPEG lossless	2.0	0.7	1.6
Lifting scheme with Haar	4.3	4.4	1.7
Gzip (Unix)	13.0	1.4	1.4

Table 2. Compression of an 1024 $\times$ 1024 integer-2 image. Platform: Sun Ultra-Sparc 250 MHz and 2 processors.