

Adaptive, Lossless Compression of 2-D Astronomical Images

C. N. Sabbey

Department of Astronomy, Yale University, New Haven, CT 06520-8101

Abstract. We describe a lossless compression program developed for incorporation into the data acquisition system of a 16 CCD drift scan survey. The program is based on switched linear prediction with Rice coding of the high order bits of the mapped prediction residuals. Several notable features of the program are its speed (≈ 5 MB/sec on a 300 MHz Pentium II), adaptivity to a variety of image types without any user-tuned parameters, and suitability for online work (it is a one-pass algorithm with minimal buffering requirements and a steady output stream).

The compression ratios and run times of the program are compared to other well-known compression programs using a variety of test images. A stand-alone version of the **encode** program is available in C from <ftp://www.astro.yale.edu/pub/sabbey/encode.tar.gz>.

1. Introduction

Although numerous programs are available for the lossy compression of astronomical images (e.g., White & Percival 1994; Starck et al. 1996), much less work has been published on lossless compression, in particular for online applications (although see Veran & Wright 1994). As a result, in producing the online software for the QUEST (QUasar Equatorial Survey Team) 16 CCD drift scan survey (Sabbey, Coppi, & Oemler 1998), we designed a custom, lossless compression program. The real-time compression provided has proved crucial for reliably acquiring and storing the nightly 30 GB of drift scan image data with the existing hardware system.

The program was required to be fast (i.e., compress at a rate comparable to the disk I/O), work in one-pass with minimal buffering, be easy to implement and incorporate into a complex system, allow streaming of the data to disk or tape, adapt to a variety of image types, and provide compression ratios comparable to more sophisticated offline programs without requiring any user-tuned parameters.

To accomplish this, we started with the well-known technique (Rice 1991) of Rice coding (Rice & Plaunt 1971) the difference between each pixel value and its linear prediction (a linear combination of the already-transmitted, neighboring pixel values). More specifically, the prediction residuals are partitioned into the N low order “noise” bits, which cannot be compressed, and the remaining $(16-N)$ high order bits, which form a sharply peaked (Laplacian) distribution, and can be efficiently transmitted as Rice codes. Rice coding is a form of entropy coding

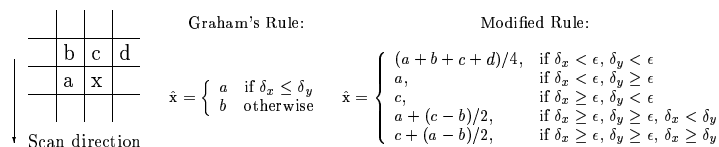


Figure 1. A prediction, \hat{x} , is formed for the current pixel, x , using one of several possible linear combinations of the neighboring pixels (a , b , c , and d). A modification of Graham's rule is used to select one of the linear combinations based on a comparison of the local gradient approximations ($\delta_x = |c - b|$ and $\delta_y = |a - b|$) to a noise threshold ϵ .

(in which more frequently occurring values are assigned shorter codewords) that is less optimal but faster than Huffman coding.

To extend this technique of Rice coding the most significant bits of the differential signal, we investigated the selection of the linear predictor, the calculation of the partition size N , and how to make both adaptive at each pixel of the image in a fast and automated way without transmitting overhead information. The adaptivity is crucial for obtaining consistently high compression ratios and eliminating parameters that require user tuning. Some examples of image variation that the program should adapt to are changes in the background level, the noise level, the object surface density and shape, the image type (e.g., photometry or slitless spectroscopy), and the existence of bad CCD columns and other defects.

To adaptively select the predictor we modified Graham's rule (see Netravali & Lamb 1980, and Fig. 1) for use with astronomical images. In relatively flat image regions, the mean of several neighboring pixels will be selected as the predictor; otherwise the direction and order of a predictor will be selected based on the local correlation. For example, pixel values in a bad CCD column will be predicted using the adjacent previous row value, while a higher order predictor in the dispersion direction will be used to predict bright spectrum pixels in a slitless spectroscopy image.

An alternative approach to making the prediction adaptive is to subdivide the image into blocks (e.g., 8×8). Then for each block all the predictors can be tested and the most effective one used. However, for our image set the optimal block size and shape varied among the images and within a given image. Also, the blocks were not aligned with the image features, and there was a conflict between using small blocks to allow rapid adaptivity but needing large blocks to limit the overhead of indicating the predictor used. Due to these and other obstacles (e.g., the increased computation required), we resorted to the modified Graham's rule described above.

The partition size (number of "noise" bits $N = N_s + N_r$) is adaptively selected at each pixel from the sum of a slowly varying term, N_s , which accounts for the background noise level, and a rapidly varying term, N_r , which accounts for structure remaining in the prediction residual image. N_s for a given image block is obtained using the number of 1-bit Rice codes transmitted in the previous image block (N_s is adjusted to keep the probability of Rice codeword 0 near

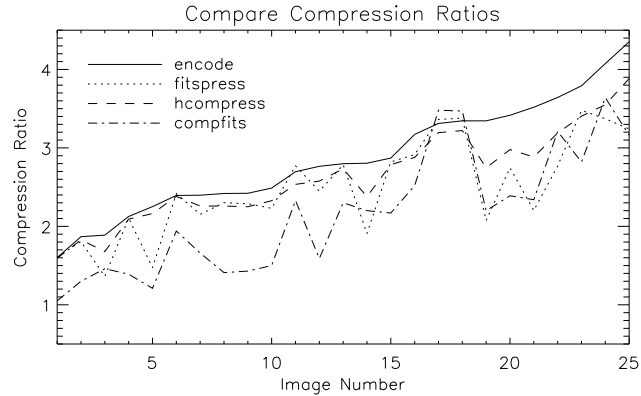


Figure 2. The performance of **encode** is evaluated by comparison to the lossless output of three well-known compression programs. The **encode** program often provides significantly improved compression ratios (the average percent difference is 22% improvement).

optimal). N_r is obtained at each pixel from a table lookup using the prediction residual from the previous pixel as an index.

The compression ratios and run times of the **encode** program were evaluated by comparison to three well-known compression programs: **hcompress**, **fitspress**, and **compfits+compact**. The test images used are listed in Table 1, and the results are presented in Figs. 2 and 3. Although the three comparison programs have important uses beyond lossless compression (e.g., **hcompress** and **fitspress** are often used for lossy compression), they nonetheless represent the state of the art and provide a relevant standard. Further details of the **encode** algorithm can be found in Sabbey et al. (1998).

References

- Netravali, A. N. & Limb, J. O. 1980, Proc. IEEE, 68, 366
- Rice, R. F. 1991, Some Practical Universal Noiseless Coding Techniques, Part III, Module PSI14,K+ (JPL Publ. 91-3), (Pasadena: JPL)
- Rice, R. F. & Plaunt, J. R. 1971, IEEE Trans. Comm. Tech., 19, 889
- Sabbey, C. N., Coppi, P., & Oemler, A. 1998, PASP, 110, 1067
- Starck J.-L., Murtagh, F., Pirenne, B., & Albrecht, M. 1996, PASP, 108, 446
- Veran, J. P. & Wright J. R. 1994, in ASP Conf. Ser., Vol. 61, Astronomical Data Analysis Software and Systems III, ed. D. R. Crabtree, R. J. Hanisch, & J. Barnes (San Francisco: ASP), 519
- White, R. L. & Percival, J. W. 1994, in SPIE Proc., Vol. 2199, Advanced Technology Optical Telescopes V, ed. L. M. Stepp (Bellingham: SPIE), 703

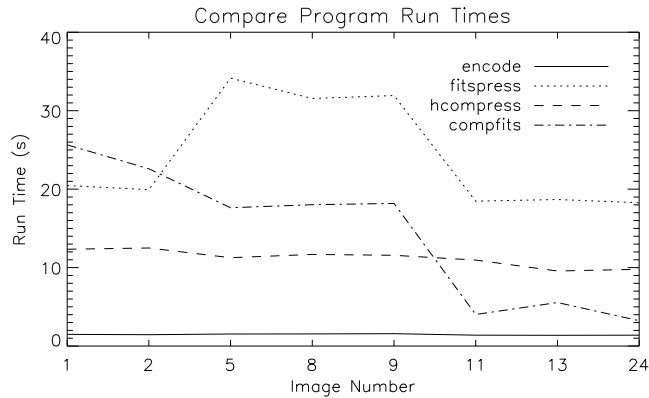


Figure 3. The speed of **encode** is evaluated by comparison to three well-known compression programs (using a 300 MHz Pentium II). Only the large (≈ 8 MB) images were used to avoid measuring times $\ll 1$ s and to use a consistent image size. The **encode** program is found to be an order of magnitude faster in many cases.

Table 1. 25 test images used to evaluate the **encode** program.

No.	Name	Size	Description	Bkg Noise
1	simn200	2048 \times 2048	IRAF simulated stars/galaxies	200.0
2	prism	2048 \times 2048	objective prism slitless spectroscopy	70.0
3	ngc0001	321 \times 507	NGC 3210 in V, globular cluster	45.5
4	com0001	500 \times 500	halo of NGC 4874, strong image gradients	50.0
5	flat	2176 \times 2048	Flatfield image, tapered edges	40.0
6	noise	1024 \times 1024	Simulated noise image, flat background	20.0
7	gal0002	264 \times 427	galaxy cluster (coadded)	24.7
8	grism2	2176 \times 2048	grism slitless spectroscopy	26.0
9	grism	2176 \times 2048	grism slitless spectroscopy	25.0
10	m57	1080 \times 1080	M57 Ring Nebula, strong image gradients	10.0
11	simusamp	2048 \times 2048	IRAF simulated undersampled stars/galaxies	10.0
12	fiber	100 \times 512	IRAF simulated multifiber spectra image	3.0
13	simosamp	2048 \times 2048	IRAF simulated oversampled stars/galaxies	10.0
14	ngc0002	322 \times 508	NGC 3210 in V, shorter exposure	7.5
15	gal0001	320 \times 503	galaxy cluster 0637-53	9.7
16	sgp0001	234 \times 441	deep SGP in J	7.7
17	gal0004	1000 \times 1000	SA 68	5.9
18	gal0003	1000 \times 1000	SA 68	6.0
19	tuc0003	420 \times 246	47 Tuc off-center in V	8.6
20	for0001	305 \times 512	Fornax dwarf in V, crowded star field	8.6
21	tuc0004	420 \times 246	47 Tuc off-center in B	6.3
22	hst	800 \times 800	HST image of globular cluster, crowded	2.0
23	sgp0002	234 \times 441	deep SGP in R	4.0
24	simn2	2048 \times 2048	IRAF simulated stars/galaxies	2.0
25	for0002	305 \times 497	Fornax dwarf in B, crowded star field	4.8

Of the 25 test images used to evaluate the **encode** program, 13 comprise the “standard test images” suite used at the Two-Dimensional Photometry Systems and Test Images Workshop (ESO/ECF 1989). The remaining 12 images include simulated and observed photometry and spectroscopy image data with varying noise levels, crowding, etc.