An Extension of FITS for Data Compression

Archibald Warnock III [1]
Robert S. Hill [1]
Barbara B. Pfarr [1]
and
D. C. Wells [2]
September 25, 1991

Abstract

An extension of the FITS format to allow for data compression is presented. This extension will allow a variety of algorithms to be utilized to reduce the storage requirements for images and ASCII data. Simple rules permit the complete reconstruction of the entire original FITS byte stream, yet the header for the compressed data conforms to the existing Generalized FITS Extension Agreement. Optional keywords permit description of the compressed data without requiring decompression. An algorithm for one compression scheme (Previous Pixel) is presented as an example.

Introduction

The ever-increasing use of digital detectors has expanded the amount of data available to astronomers beyond the point of easy distribution. Data rates on existing computer networks are too slow to permit the flow of large amounts of data. Even the volume of tabular data in digital form can stretch the capabilities of distribution channels.

The Large-Scale Phenomena Network (LSPN) of the International Halley Watch (IHW) has had the task of digitizing wide-field photographic plates of Comet P/Halley. The physical size of the plates and the high spatial resolution result in extremely large digital images, often as large as 4096 lines by 4096 samples of 10-bit numbers (*i.e.*, individual images may require as much as 32 megabytes of storage).

Production images are 12 megabytes in size, on average, and the LSPN has generated about 1500 of them. The IHW Digital Archive could be expected to consist of perhaps 35 to 40 CD-ROM disks, most of them containing only images. By contrast, an archive containing only compressed images would have on the order of 20 disks, with a corresponding decrease in cost. It is essential that there be some portable data compression scheme to reduce the physical storage requirements of the imagery, and thereby reduce the mastering and production costs for a complete archive on CD-ROM. The expense and effort associated with distributing the complete archive of digital data in uncompressed form on any medium would be prohibitive.

[1]ST Systems Corp., NASA/Goddard Space Flight Center, Greenbelt, MD 20771
[2]National Radio Astronomy Observatory, Edgemont Road, Charlottesville, VA 22903

The Guide Star Catalog from the Space Telescope Science Institute requires over 1 gigabyte of storage just for its tabular data. The imagery from which the catalog was extracted requires over 400 gigabytes. The Wide-Field Camera of Space Telescope will generate images with 1600 pixels on a side. The distribution of such vast amounts of data is expensive in both time and resources.

For some time, the microcomputer community has had *de facto* standards for the interchange and reconstruction of compressed data (programs such as ZIP, ARC and ZOO). These programs serve to reduce the expense associated with acquiring data and to allow wider access to files of interest by reducing the time required for file transfer via telephone or network. We propose a similar scheme for the interchange of compressed data within the framework of the Generalized FITS Extension Agreement.

The FITS Header for Compressed Data

A compressed file can be considered as an extension to basic FITS in the following sense: the compressed data can be read simply as a byte stream. It is fully documented by an extension header written according to the FITS prescription. Thus, one retains the capability of examining the header of the compressed data with a FITS reader. The end result (after decompression) is again a standard FITS file, in the sense that the data is restored to its original uncompressed state, complete with a valid header.

Two types of size information are required. The basic FITS keywords (BITPIX, NAXIS, NAXIS1 – NAXISn) must continue to define the attributes of the byte stream in the form in which the user receives it. New FITS keywords then define the file as it is to be reconstructed.

We propose that the keyword XTENSION take the value 'COMPRESS' if the data has been compressed, the number of compression algorithms used be given by the value of the keyword 'COMPRES' and that a series of keywords COMPRES1, COMPRES2, ..., COMPRESn give the names of the compression schemes used on the original data, in the order in which the compressions were performed; *i.e.*, the scheme given by COMPRES1 was used first, that given by COMPRES2 was used second, and so forth. Note that the entire input FITS byte stream, header and all, is to be compressed, and a new header is affixed to the beginning of the resulting compressed byte stream.

We recommend the use of a set of optional keywords to document the format of the original data stream without requiring that any decompression be performed. They will allow, at least in the case of image data, the complete reconstruction of the original image parameters. The number of axes in the original uncompressed data is given by the keyword LAXIS (for logical axis), and the corresponding dimensions are given by the keywords LAXIS1 through LAXISn. The keyword LBITPIX will define the precision of the uncompressed data, corresponding to BITPIX after decompression. Two keywords (LEXTEND and LXTENSN) should be used if the uncompressed data was a FITS extension, and correspond to EXTEND and XTENSION, respectively. While not mandatory, these keywords make the parameters of the original data

```
First record:
0........1.........2.........3.........4.........5.........6.........7.....
12345678901234567890123456789012345678901234567890123456789012345678901...
SIMPLE  =                    T / VALID FITS FORMAT
BITPIX  =                    8 / BYTE DATA
NAXIS   =                    0 / NO DATA RECORDS YET
EXTEND  =                    T / THERE MAY BE EXTENSION RECORDS
END

Extension record:
0........1.........2.........3.........4.........5.........6.........7.....
12345678901234567890123456789012345678901234567890123456789012345678901...
XTENSION= 'COMPRESS'           / THE DATA HAS BEEN COMPRESSED
BITPIX  =                    8 / BYTE DATA ONLY
NAXIS   =                    1 / THE DATA IS JUST A BYTE STREAM
NAXIS1  =                12345 / OF THIS MANY BYTES
PCOUNT  =                    0 / NO PARAMETERS PRECEEDING THE DATA
GCOUNT  =                    1 / ONLY ONE GROUP
LAXIS   =                    2 / NUMBER OF AXES, UNCOMPRESSED IMAGE
LAXIS1  =                  512 / LOGICAL NAXIS1, UNCOMPRESSED IMAGE
LAXIS2  =                  512 / LOGICAL NAXIS2, UNCOMPRESSED IMAGE
LBITPIX =                   16 / LOGICAL BITS PER PIXEL
LEXTEND =                    F / UNCOMPRESSED DATA NOT AN EXTENSION
COMPRES =                    2 / NUMBER OF COMPRESSION STEPS
COMPRES1= 'PREVPIXEL'          / FIRST COMPRESSION SCHEME USED
COMPRES2= 'HUFFMAN  '          / SECOND COMPRESSION SCHEME USED
END
```

Figure 1: Sample compression header

accessible, and so are strongly recommended.

Figure 1 illustrates the syntax for the proposed compressed file extension. This scheme identifies the data as having been compressed (XTENSION = 'COMPRESS') and identifies the compression scheme(s) used (COMPRES1 = 'PREVPIXEL', COMPRES2 = 'HUFFMAN') in the header, so that any FITS reader can decide whether or not it knows how to handle the data. It defines the real length of the data file (NAXIS1 = 12345), so that optionally the data can be skipped. It also preserves the dimensions of the original data structure as "logical" attributes (LAXIS, LAXISn, etc.), so that the format of the input FITS file can be recreated without decompressing the data.

Any keywords which must accompany the uncompressed data could be put into the extension header after the mandatory and recommended ones, and could be copied to the output

header when the file is decompressed, if desired.

Possible Compression Schemes

The IHW restricted its consideration of compression schemes to those which preserve the full dynamic range of the data, in order to preserve the maximum possible photometric accuracy. Of these, the so-called "instantaneous" methods were preferred; that is, those methods in which the next output number is computed from the current input number and a small set of status variables.

The advantage of instantaneous decompression schemes over others is that no scratch space is required to hold the uncompressed file, since the decompression takes place "on the fly." Thus, compression can have a minimal effect on user interfaces. A compressed file in FITS format, with appropriate keywords, can be read directly from the distribution medium in virtually the same way that uncompressed files are currently read, with the decompression being performed as the records are read and unpacked. This is an important consideration for users of large compressed data archives, such as the IHW CD-ROM archive. Although the actual compression may require substantial resources, it is done infrequently, in general; the corresponding decompression can require substantially fewer resources.

The IHW believes that the Previous Pixel (or first-differences) algorithm is the one giving the best compression ratio for the least computational effort and it has been adopted for use on the CD-ROM archive disks[1]. Details of the algorithm as implemented by the IHW are in the Appendix. For the images digitized by the LSPN on the PDS 1010A Microdensitometer at Goddard Space Flight Center, the compressed files are typically between 52% and 55% the size of the original uncompressed data. Appendix B discusses some additional compression algorithms.

Discussion

One of the reasons for suggesting keywords whose values explicitly state the compression schemes used is precisely because no single scheme is best under all circumstances. Our approach allows flexibility in selecting any appropriate (or even inappropriate) compression method and specifying it unambiguously, so long as the algorithm has been published and agreed upon.

[1]Note: The compressed files on the IHW CD-ROMs differ slightly from the format proposed here because the production of the IHW compressed files was based on a preliminary version of this proposal. The major difference in the compression used by the IHW is that only the images were compressed, not the headers, and the entire original FITS header is embedded in the Extension header. There is no header data in the compressed byte stream itself.

Although the best compression for the actual data may not be optimal for the accompanying header, if the data take up enough space, the relative penalty for using the same scheme on the header will be small.

New compression schemes should be added to the recognized list in the same way that FITS extensions have been proposed and adopted in the past (Grosbøl *et al* 1988, Harten *et al* 1988). The International Halley Watch has inaugurated this procedure with the Previous Pixel algorithm and will publish source code and pseudo-code for decompression both on the archive CD-ROM and in print.

It is also possible to interpret compression like blocking; *i.e.*, as irrelevant to the logical structure of the data. Although this is philosophically true, we remain convinced of the need to identify and support the most useful practical options, as is already being done for World Coordinate systems. Agreement on a flexible set of keywords will allow new algorithms to be evaluated and added later, in full accordance with any current agreement. Although it is possible to imitate the personal computer approach by writing code that analyzes a given input image and selects the best compression scheme, it is impossible to anticipate all the schemes that might be desirable in the future.

Another alternative to the current proposal would be to use a standalone program to compress both header and data, but without attaching a new set of FITS headers. In such a case, the header records would be unreadable to existing FITS software until after decompression. It would be wasteful of both time (for decompression) and disk space (for scratch storage) and, we believe, contrary to the basic FITS philosophy of allowing examination of the header before deciding what to do with the data.

Nothing in this proposal precludes the use of a standalone decompressor. It is certainly possible to skip over the header to reach the data and decompress the data separately. Note also that nothing in this proposal restricts compression to images. The scheme specified here would work equally well with ASCII data, as would it with any FITS extension, such as tables.

The Planetary Data System has already used compression in the production of the Voyager CD-ROM disks, although the descriptors are decidedly not FITS. The IHW has already performed the necessary prototyping for the Previous Pixel compression algorithm. For the IHW, compressed images are an absolute necessity (or perhaps a necessary evil).

Not only is our approach independent of the storage medium, but also, it avoids reliance on coded filename extensions and the like. Nevertheless, data analysis packages can easily recognize compressed data and either process or skip, according to their capabilities.

Data compression has proven its value in many areas — PC-based telecommunications and file transfer, transmission of spacecraft data, etc. It is only a matter of time until astronomers appreciate how attractive it can be for images in general (consider the prospect of 2048 by 2048 CCD chips with three bytes per pixel). Now is a good time to agree on how to incorporate compression into the general data interchange standards so that we can avoid revisions in the future.

Bibliography

Grosbøl, P., Harten, R. H., Greisen, E. W., and Wells, D. C., 1988, Astron. Astrophys. Suppl. Ser., **73**, 359.
Harten, R. H., Grosbøl, P., Greisen, E. W., and Wells, D. C., 1988, Astron. Astrophys. Suppl. Ser., **73**, 365.

Authors

Archibald Warnock III
ST Systems Corp.
Code 681
NASA/Goddard Space Flight Center
Greenbelt, MD 20771
(301)286-3965
SPAN: STARS::WARNOCK or 6168::WARNOCK
Internet: warnock@stars.gsfc.nasa.gov
FAX: (301)286-8709

Robert S. Hill
ST Systems Corp.
Code 681
NASA/Goddard Space Flight Center
Greenbelt, MD 20771

Barbara B. Pfarr
ST Systems Corp.
Code 681
NASA/Goddard Space Flight Center
Greenbelt, MD 20771

Don C. Wells
National Radio Astronomy Observatory
Edgemont Road
Charlottesville, VA 22901
Internet: dwells@nrao.edu

Appendix A — Previous Pixel Compression

The algorithm for the Previous Pixel compression scheme to be used by the International Halley Watch is given here in pseudocode. It is based on the observation that, for many images of 16-bit data, the pixel-to-pixel differences may often be coded within the dynamic range available in 8 bits, yielding a substantial savings in file size from a small computational investment.

All differences which lie in the range [-127,127] can be coded in a single byte. Each difference has the bias value 127 added to it in order to avoid problems on machines which require unsigned byte data. This yields the unsigned range [0,254] (or [0,*FE*] *hex*) in the actual data stream.

The value 255 (or *FF hex*) is reserved as a flag to indicate that the difference between the two current pixels is too large to be stored in 8 bits. In this case, the two bytes that follow the flag byte are to be interpreted as a new 16-bit pixel value, which then provides a new zero-point for the differences.

No allowances are made for ends of lines; that is, the successive differences are allowed to cross from the right-hand edge of the image to the next line at the left hand edge. For images with smooth backgrounds, this will often result in another 8-bit difference, and so save a few more bytes. Note that the number of samples in a line is given by the keyword LAXIS2, so that there is no need to flag the start of a new line. Note also that the resulting compressed byte stream is not unique. A 255 flag and a new 16-bit pixel value may be inserted at any point in the byte stream and the byte stream will still uncompress uniquely. This allows

considerable flexibility in buffering the original input image.

PREVIOUS PIXEL ALGORITHM

COMPRESSION:

Load 255 into output record
READ first record
Set first value to be *PREVPIXEL*
Load first value into output record (using FITS-required byte ordering)
DO UNTIL no more data on input
    IF input buffer is empty **THEN** read next record
    Compute difference between *CURRENTPIXEL* and *PREVPIXEL*
    IF this difference is within −127:127 **THEN**
        Add bias of 127 to the difference
        Convert the biased difference to a single byte
        Load value of difference byte into output record
        IF output record full, **WRITE** out record
    ELSE IF the difference is outside −127:127 **THEN**
        Load flag byte (255) into output record
        IF output record full, **WRITE** out record
        Load high byte of *CURRENTPIXEL* into output record
        IF output record full, **WRITE** out record
        Load low byte of *CURRENTPIXEL* into output record
        IF output record full, **WRITE** out record
    ENDIF
    Set *PREVPIXEL* equal to *CURRENTPIXEL*
ENDDO
IF partially filled output buffer remains **THEN**
    Set remainder the buffer to NULL
    WRITE out final (partial) record
ENDIF

DECOMPRESSION:

READ first record
Verify that first byte is 255
Set first value to be *PREVPIXEL*
Load first value into output record
DO UNTIL no more data on input
    IF input buffer is empty **THEN** read next record
    Get *CURRENTBYTE* from input buffer
    IF *CURRENTBYTE* is not equal to 255 (is a difference) **THEN**
        NEWPIXEL = *PREVPIXEL* + *CURRENTBYTE* − 127
        Load *NEWPIXEL* into output record
        IF output record full, **WRITE** out record
    ELSE IF *CURRENTBYTE* = 255 **THEN**
        IF input buffer has < 2 bytes left **THEN** read next record
        Set *NEWPIXEL* to the next 2 bytes in the input buffer
        Load *NEWPIXEL* into output record
        IF output record full, **WRITE** out record
    ENDIF
    Set *PREVPIXEL* equal to *NEWPIXEL*
ENDDO
IF partially filled output buffer remains **THEN**
    Blank to the end of the buffer
    WRITE out final (partial) record
ENDIF

Appendix B — Other Compression Schemes

Table 1 contains a listing of compressed data sizes for various FITS format files. The compression has been performed on an IBM-PC/AT compatible machine running MS-DOS. The programs used are all generally available compression programs distributed under the "Shareware" concept. They are presented here to give some idea of the "state-of-the-art" in file compression. None of these programs or algorithms are included in the current proposal, although the algorithms have been well tested and could easily be implemented under the conventions presented here.

The **SQ**[1] program uses a Huffman coding technique. The programs **PAK**[2], **PKPAK**[3] and **ZIP**[4] all use modifications of the Lempel-Zev or Lempel-Zev-Welch compression algorithm. The **LHARC**[5] program uses an adaptive Huffman coding with LZSS encoding.

The first sixteen files (FITS001 through FITS016) in Table 1 are from the FITS Test Tape, version 4, available from the National Radio Astronomy Observatory. The remaining images are:

30 Doradus - Image of 30 Doradus from the Guide Star Catalog (GSC Sampler CD-ROM from the Space Telescope Science Institute, digitized at 512×512×16-bit: lots of stars and nebulosity.

---

[1] **SQPC** version 1.31, ©1986 by Vernon D. Buerg, 456 Lakeshire Dr, Daly City, CA 94015

[2] **PAK** version 1.0, ©1988 by NoGate Consulting, P. O. Box 88115, Grand Rapids, MI 49518

[3] **PKPAK** version 3.61, ©1986-1988 by PKWare, Inc., 7545 N. Port Washington Rd., Glendale, WI 53217

[4] **PKZIP** version 1.02, ©1989 by PKWare, Inc., 7545 N. Port Washington Rd., Glendale, WI 53217

[5] **LHARC** v1.12b, ©1988-1989 by Haruyasi Yoshizaki

η Carina - Image of η Carina from the GSC Sample CD-ROM, digitized at 512×512×16-bit: lots of stars and nebulosity.

P/G-Z - Image of Comet P/Giacobini-Zinner from the IHW P/G-Z Test CD-ROM, digitized at 2048×2048×10-bit: very smooth background and small stars.

NGC 300 - Image of NGC 300 from the GSC Sampler CD-ROM, digitized at 800×800×16-bit: lots of stars.

NGC 362 - Image of NGC 362 from the GSC Sampler CD-ROM, digitized at 512×512×16-bit: lots of stars in globular cluster.

Input File	Compression Percentage By Program					
	FITS	PKPAK	LHARC	PAK	SQ	ZIP
FITS001	100.00	12.26	10.68	12.50	17.05	11.06
FITS002	100.00	12.87	11.46	12.98	17.47	11.97
FITS003	100.00	84.47	71.15	80.20	76.50	89.95
FITS004	100.00	74.67	64.05	65.10	79.31	67.10
FITS005	100.00	21.88	19.94	20.92	28.85	17.25
FITS006	100.00	97.33	92.91	95.00	97.32	99.56
FITS007	100.00	65.63	58.78	59.55	70.55	63.06
FITS008	100.00	79.59	70.44	70.56	72.30	76.76
FITS009	100.00	65.46	51.50	55.70	68.61	54.24
FITS010	100.00	77.37	69.88	76.90	72.25	74.75
FITS011	100.00	14.64	15.37	14.80	22.33	17.82
FITS012	100.00	42.21	41.18	39.49	66.60	42.99
FITS013	100.00	58.56	50.22	50.98	65.07	53.18
FITS014	100.00	17.96	18.44	17.04	35.72	18.63
FITS015	100.00	46.65	47.03	44.03	62.27	50.30
FITS016	100.00	48.39	40.61	38.77	40.05	41.96
30 Doradus	100.00	84.55	84.17	91.08	84.58	95.08
η Carina	100.00	88.85	86.54	93.56	88.89	97.43
P/G-Z	100.00	62.20	62.52	62.38	81.99	66.10
NGC 300	100.00	85.89	83.70	90.76	85.95	93.40
NGC 362	100.00	81.15	81.73	89.51	81.20	90.79

Table 1: Compression Ratios for Different Algorithms. The values given are the ratios of the compressed file size to the original file size, given as a percentage.