High-Performance Compression of Astronomical Images

Richard L. White

Joint Institute for Laboratory Astrophysics, University of Colorado
Campus Box 440, Boulder, CO 80309
and
Space Telescope Science Institute
3700 San Martin Drive, Baltimore, MD 21218

rlw@stsci.edu

Summary

Astronomical images have some rather unusual characteristics that make many existing image compression techniques either ineffective or inapplicable. A typical image consists of a nearly flat background sprinkled with point sources and occasional extended sources. The images are often noisy, so that lossless compression does not work very well; furthermore, the images are usually subjected to stringent quantitative analysis, so any lossy compression method must be proven not to discard useful information, but must instead discard only the noise. Finally, the images can be extremely large. For example, the Space Telescope Science Institute has digitized photographic plates covering the entire sky, generating 1500 images each having 14000×14000 16-bit pixels. Several astronomical groups are now constructing cameras with mosaics of large CCDs (each 2048×2048 or larger); these instruments will be used in projects that generate data at a rate exceeding 100 MBytes every 5 minutes for many years.

An effective technique for image compression may be based on the H-transform (Fritze *et al.* 1977). The method that we have developed can be used for either lossless or lossy compression. The digitized sky survey images can be compressed by at least a factor of 10 with no noticeable losses in the astrometric and photometric properties of the compressed images. The method has been designed to be computationally efficient: compression or decompression of a 512×512 image requires only 4 seconds on a Sun SPARCstation 1. The algorithm uses only integer arithmetic, so it is completely reversible in its lossless mode, and it could easily be implemented in hardware for space applications.

1. Introduction

Astronomical images consist largely of empty sky. Compression of such images can reduce the volume of data that it is necessary to store (an important consideration for large scale digital sky surveys) and can shorten the time required to transmit images (useful for remote observing or remote access to data archives.)

Data compression methods can be classified as either "lossless" (meaning that the original data can be reconstructed exactly from the compressed data) or "lossy" (meaning that the uncompressed image is not exactly the same as the original.) Astronomers often insist that they can accept only lossless compression, in part because of conservatism, and in part because the familiar lossy compression methods sacrifice some information that is

needed for accurate analysis of image data. However, since all astronomical images contain noise, which is inherently incompressible, lossy compression methods produce much better compression results.

A simple example may make this clear. One of the simplest data compression techniques is run-length coding, in which runs of consecutive pixels having the same value are compressed by storing the pixel value and the repetition factor. This method is used in the standard compression scheme for facsimile transmissions. Unfortunately, it is quite ineffective for lossless compression of astronomical images because even though the sky is *nearly* constant, the noise in the sky ensures that only very short runs of equal pixels occur. The obvious way to make run-length coding more effective is to force the sky to be exactly constant by setting all pixels below a threshold (chosen to be just above the sky) to the mean sky value. However, then one has lost any information about objects close to the detection limit. One has also lost information about local variations in the sky brightness, which severely limits the accuracy of photometry and astrometry on faint objects. Worse, there may be extended, low surface brightness objects that are not detectable in a single pixel but that are easily detected when the image is smoothed over a number of pixels; such faint structures are irretrievably lost when the image is thresholded to improve compression.

2. The H-transform

Fritze *et al.* (1977; see also Richter 1978 and Capaccioli *et al.* 1988) have developed a much better compression method for astronomical images based on what they call the *H-transform* of the image. A similar transform called the S-transform has also been used for image compression (Blume & Fand 1989). The H-transform is a two-dimensional generalization of the Haar transform (Haar 1910). The H-transform is calculated for an image of size $2^N \times 2^N$ as follows:

- Divide the image up into blocks of 2×2 pixels. Call the 4 pixels in a block a_{00}, a_{10}, a_{01}, and a_{11}.
- For each block compute 4 coefficients:

$$\begin{aligned} h_0 &= (a_{11} + a_{10} + a_{01} + a_{00})/2 \\ h_x &= (a_{11} + a_{10} - a_{01} - a_{00})/2 \\ h_y &= (a_{11} - a_{10} + a_{01} - a_{00})/2 \\ h_c &= (a_{11} - a_{10} - a_{01} + a_{00})/2 \end{aligned}$$

- Construct a $2^{N-1} \times 2^{N-1}$ image from the h_0 values for each 2×2 block. Divide that image up into 2×2 blocks and repeat the above calculation. Repeat this process N times, reducing the image in size by a factor of 2 at each step, until only one h_0 value remains.

This calculation can be easily inverted to recover the original image from its transform. The transform is exactly reversible using integer arithmetic if one does not divide by 2 for

the first set of coefficients. It is straightforward to extend the definition of the transform so that it can be computed for non-square images that do not have sides that are powers of 2. The H-transform can be performed in place in memory and is very fast to compute, requiring about $16M^2/3$ (integer) additions for a $M \times M$ image.

The H-transform is a simple 2-dimensional wavelet transform. It has several advantages over some other wavelet transforms that have been applied to image compression (*e.g.*, Daubechies 1988). First, the transform can be performed entirely with integer arithmetic, making it exactly reversible. Consequently it can be used for either lossless or lossy compression (as indicated below) and one does not need a special technique for the case of lossless compression (as was required, *e.g.*, , for the JPEG compression standard.)

A second major advantage is that the H-transform is a natively 2-dimensional wavelet transform. The standard 1-dimensional wavelet transforms are extended to two dimensions by transforming the image first along the rows, then along the columns. Unfortunately, this generates many wavelet coefficients that are high frequency (hence localized) in the x-direction but low frequency (hence global) in the y-direction. Such coefficients are counter to the philosophy of the wavelet transform: high-frequency basis functions should be confined to a relatively small area of the image. Discarding these mixed-scale terms, which may be negligible compared to the noise, generates very objectionable artifacts around point sources and edges in the image. The H-transform, on the other hand, is a fully 2-dimensional wavelet transform, with all high frequency terms being completely localized. It is consequently more suitable for image compression and produces fewer artifacts.

A possible disadvantage of the H-transform is that other wavelet transforms take better advantage of the continuity of pixel values within images, so that they can produce higher compressions for very smooth images. However, for astromical images (which are mostly flat sky sprinkled with point sources) the smoothness built into higher-order transforms can actually reduce the effectiveness of compression, because one must keep more coefficients to describe each point source.

3. Compression Using the H-transform

If the image is nearly noiseless, the H-transform is somewhat easier to compress than the original image because the differences of adjacent pixels (as computed in the H-transform) tend to be smaller than the original pixel values for smooth images. Consequently fewer bits are required to store the values of the H-transform coefficients than are required for the original image. For very smooth images the pixel values may be constant over large regions, leading to transform coefficients that are zero over large areas.

Noisy images still do not compress well when transformed, though. Suppose there is noise σ in each pixel of the original image. Then from simple propagation of errors, the noise in each of the H-transform coefficients is also σ. To compress noisy images, divide each coefficient by $S\sigma$, where $S \sim 1$ is chosen according to how much loss is acceptable. This reduces the noise in the transform to $0.5/S$, so that large portions of the transform are zero (or nearly zero) and the transform is highly compressible.

Why is this better than simply thresholding the original image? As discussed above, if we simply divide the image by σ then we lose all information on objects that are within

1σ of sky in a *single* pixel, but that are detectable by averaging a *block* of pixels. On the other hand, in dividing the H-transform by σ, we preserve the information on any object that is detectable by summing a block of pixels! The quantized H-transform preserves the mean of the image for every block of pixels having a mean significantly different than that of neighboring blocks of pixels.

As an example, Figure 1 shows a 128×128 section (3.6×3.6 arcmin) from a digitized version of the Palomar Observatory–National Geographic Society Sky Survey plate containing the Coma cluster of galaxies. Figures 2, 3, and 4 show the resulting image for $S \simeq 0.5$, 1, and 2. These images are compressed by factors of 10, 20, and 60 using the coding scheme described below. In all cases a logarithmic gray scale is used to show the maximum detail in the image near the sky background level; the noise is clearly visible in Figure 1. The image compressed by a factor of 10 is hardly distinguishable from the original. In quantizing the H-transform we have adaptively filtered the original image by discarding information on some scales and keeping information on other scales. This adaptive filtering is most apparent for high compression factors (Fig. 4), where the sky has been smoothed over large areas while the images of stars have hardly been affected.

The adaptive filtering is, in itself, of considerable interest as an analytical tool for images (Capaccioli *et al.* 1988). For example, one can use the adaptive smoothing of the H-transform to smooth the sky without affecting objects detected above the (locally determined) sky; then an accurate sky value can be determined by reference to any nearby pixel.

The blockiness that is visible in Figure 4 is the result of difference coefficients being set to zero over large areas, so that blocks of pixels are replaced by their averages. It is possible to eliminate the blocks by an appropriate filtering of the image. A simple but effective filter can be derived by simply adjusting the H-transform coefficients as the transform is inverted to produce a smooth image; as long as changes in the coefficients are limited to $\pm S\sigma/2$, the resulting image will still be consistent with the thresholded H-transform.

4. Efficient Coding

The quantized H-transform has a rather peculiar structure. Not only are large areas of the transform image zero, but the non-zero values are strongly concentrated in the lower-order coefficients. The best approach we have found to code the coefficient values efficiently is quadtree coding of each bitplane of the transform array. Quadtree coding has been used for many purposes (see Samet 1984 for a review); the particular form we are using was suggested by Huang and Bijaoui (1991) for image compression.

- Divide the bitplane up into 4 quadrants. For each quadrant code a '1' if there are any 1-bits in the quadrant, else code a '0'.
- Subdivide each quadrant that is not all zero into 4 more pieces and code them similarly. Continue until one is down to the level of individual pixels.

This coding (which Huang and Bijauoi call "hierarchic 4-bit one" coding) is obviously very well suited to the H-transform image because successively lower orders of the H-transform coefficients are located in successively divided quadrants of the image.

We follow the quadtree coding with a fixed Huffman coding that uses 3 bits for quadtree values that are common (*e.g.*, 0001, 0010, 0100, and 1000) and uses 4 or 5 bits for less common values. This reduces the final compressed file size by about 10% at little computational cost. Slightly better compression can be achieved by following quadtree coding with arithmetic coding (Witten, Bell, and Cleary 1987), but the CPU costs of arithmetic coding are not, in our view, justified for 3–4% better compression. We have also tried using arithmetic coding directly on the H-transform, with various contexts of neighboring pixels, but find it to be both computationally inefficient and not significantly better than quadtree coding.

For completely random bitplanes, quadtree coding can actually use more storage than simply writing the bitplane directly; in that case we just dump the bitplane with no coding.

Note that by coding the transform one bitplane at a time, the compressed data can be viewed as an incremental description of the image. One can initially transmit a crude representation of the image using only the small amount of data that is required for the sparsely populated, most significant bit planes. Then the lower bit planes can be added one by one until the desired accuracy is required. This could be useful, for example, if the data is to be retrieved from a remote database — one could examine the crude version of the image (retrieved very quickly) and abort the transmission of the rest of the data if the image is judged to be uninteresting.

5. Astrometric and Photometric Properties of Compressed Images

Astronomical images are not simply subjected to visual examination, but are also subjected to careful quantitative analysis. For example, for the image in Figure 1 one would typically like to do astrometric (positional) measurements of objects to an accuracy much better than 1 pixel, photometric (brightness) measurements of objects to an accuracy limited only by the detector response and the noise, and accurate measurements of the surface brightness of extended sources.

We have done some experiments to study the degradation of astrometry and photometry on the compressed images compared to the original images (White, Postman, and Lattanzi 1991). Even the most highly compressed images have very good photometric properties for both point sources and extended sources; indeed, photometry of extended objects can be improved by the adaptive filtering of the H-transform (Cappacioli *et al.* 1988). Astrometry is hardly affected by the compression for modest compression factors (up to about a factor of 20 for our digitized photographic plates), but does begin to degrade for the most highly compressed images.

These results are based on tests carried out with tools optimized for the original images; it is likely the best results will be obtained for highly compressed images only with analysis tools specifically adapted to the peculiar noise characteristics of the compressed images.

6. Conclusions

In order to construct the Guide Star Catalog for use in pointing the Hubble Space Telescope, the Space Telescope Science Institute scanned and digitized wide-field photographic

plates covering the entire sky. The digitized plates are of great utility, but to date it has been impossible to distribute the scans because of the massive volume of data involved (a total of about 600 Gbytes). Using the compression techniques described in this paper, we plan to distribute our digital sky survey on CD-ROMs; about 100 CD-ROMs will be required if the survey is compressed by a factor of 10.

The algorithm described in this paper has been shown to be capable of producing highly compressed images that are very faithful to the original. Algorithms designed to work on the original images can give comparable results on object detection, astrometry, and photometry when applied to the images compressed by a factor of 10 or possibly more. Further experiments will determine more precisely just what errors are introduced in the compressed data; it is possible that certain kinds of analysis will give more accurate results on the compressed data than on the original because of the adaptive filtering of the H-transform (Capaccioli *et al.* 1988).

This compression algorithm can be applied to any image, not just to digitized photographic plates. Experiments on CCD images indicate that lossless compression factors of 3–30 can be achieved depending on the CCD characteristics (*e.g.*, the readout noise). A slightly modified algorithm customized to the noise characteristics of the CCD will do better. This application will be explored in detail in the future.

We gratefully acknowledge grant from NAGW-2166 from the Science Operations Branch of NASA headquarters which supported this work. The Space Telescope Science Institute is operated by AURA with funding from NASA and ESA.

References

Blume, H., and Fand, A. 1989, *SPIE Vol. 1091, Medical Imaging III: Image Capture and Display*, p. 2.

Capaccioli, M., Held, E. V., Lorenz, H., Richter, G. M., and Ziener, R. 1988, *Astronomische Nachrichten*, **309**, 69.

Daubechies, I. 1988, *Comm. Pure and Appl. Math.*, **41**, 909.

Fritze, K., Lange, M., Möstl, G., Oleak, H., and Richter, G. M. 1977, *Astronomische Nachrichten*, **298**, 189.

Haar, A. 1910, *Math. Ann.* **69**, 331.

Huang, L, and Bijaoui, A. 1991, *Experimental Astronomy*, **1**, 311.

Richter, G. M. 1978, *Astronomische Nachrichten*, **299**, 283.

Samet, H. 1984, *ACM Computing Surveys*, **16**, 187.

White, R. L., Postman, M., and Lattanzi, M. 1991, in *Proceedings of the Edinburgh Meeting on Digital Sky Surveys*, in press.

Witten, I. H., Radford, M. N., and Cleary, J. G. 1987, *Communications of the ACM*, **30**, 520.