COMPRESSION OF GRAYSCALE SCIENTIFIC AND MEDICAL IMAGE DATA

F Murtagh[1*], M Louys[2,3], J.L. Starck[4] and F Bonnarel[2]

1 School of Computer Science, Queen's University Belfast, Belfast BT7 1NN*
Email: f.murtagh@qub.ac.uk
2 Astronomical Observatory, Rue de l'Université, 67000 Strasbourg, France
3 Laboratoire des Sciences de l'Informatique, de l'Image et de la Télédétection, ENSPS, Université Louis Pasteur, F-67000 Strasbourg, France
4 DAPNIA/SEI-SAP, CEA-Saclay, F-91191 Gif-sur-Yvette Cedex, France

ABSTRACT

A review of issues in image compression is presented, with a strong focus on the wavelet transform and other closely related multiresolution transforms. The roles of information content, resolution scale, and image capture noise, are discussed. Experimental and practical results are reviewed.

Keywords: lossy and lossless compression, wavelet transform, multiresolution transform, noise model

1 INTRODUCTION

We begin by reviewing the general principles which are used in compression, including the following. Why the wavelet transform? How are computational speed-ups obtained? What motivates one encoding approach over another? Can other non-wavelet multiscale transforms be used?

A very common property of data which we can seek to exploit in order to achieve good compression rates is a lack of variation between adjacent data values. This principle in exploited in run-length encoding, where the data value is coded along with the number of successive identical values. When there is a background trend or seasonality, taking the difference between adjacent values may result is a less changeable data stream. Pushing the latter principle further implies that smoothing is usually beneficial (if we can allow ourselves the luxury of doing that to our data). If we accept some amount of information loss in compressing our data, then identity of adjacent values can be replaced by similar adjacent values. Such similarity or correlation can be brought about by the use of some appropriate transform. The discrete cosine transform is one such transform, used in the JPEG image storage format which incorporates compression. The wavelet transform makes use of a wavelet function which is defined in both space and frequency, and in both of these dimensions it too allows for "like with like" to be moved together in transform space. The wavelet transform has good energy compacting properties, which means that in practice values created in transform space are made larger or smaller (often zero in the latter case) than in direct space.

The wavelet transform is used in the JPEG2000 standard for still images. This standard uses as dyadic discrete wavelet transforms either the Le Gall 5/3 taps reversible transform, or the Daubechies 9/7 taps non-reversible transform. A useful review of wavelet-based compression, including code, can be found in Welstead (1999). The lifting scheme, developed by Sweldens, is a flexible implementation technique for wavelet transforms. Of particular importance for

compression is the property of the lifting scheme that allows wavelet coefficients to be maintained as integers for integer inputs. This has storage and computational benefits.

Other encoding stages include entropy coding, which includes arithmetic or Huffman coding. The latter replaces data values with a variable-length code, to advantageously use a short-length code for common data values, and a longer-length code for uncommon data values. The overall entropy (average information, $-\Sigma p \log p$ where p is the probability of occurrence of the data value) is thereby minimized.

In scalar quantization, pixel values are replaced with quantized "approximants", and vector quantization takes vectors of values and replaces them with a good-fitting representative vector. In scalar quantization, assuming a Gaussian noise model for one's data, division by an integer times the standard deviation may be a useful step prior to quantizing in that it meaningfully rescales signal. Lloyd-Max (Proakis, 1995) is a well-known non-uniform quantization method (and is essentially equivalent to k-means clustering).

Given that wavelet transforms, as we have noted, render many values small or zero in transform space, it behoves us to exploit this in reading off the wavelet coefficient values. This leads to wavelet coders, and more specifically to zero trees. Targeted is the reading off of wavelet coefficients such that zero values at high wavelet scales (i.e., low resolution) are followed by zero values at progressively lower wavelet scales. Shapiro's EZW (embedded zero-tree wavelet) coding algorithm is one such algorithm, which was refined in the SPIHT (set partitioning in hierarchical trees) algorithm of Said and Pearlman. Worked examples can be found in Welstead (1999) and Xiong & Ramchandran (2000). Such embedded coding schemes draw benefit from wavelet coefficient properties, and they may allow for progressive display. Progressive transmission consists of visualizing quickly a low resolution image, and then increasing the quality with the arrival of new bits.

In work reported on below, the following lossy compression approach was used: (i) wavelet or other multiresolution transform, (ii) quantization which gives rise to the loss of information, and (iii) Huffman coding. A major benefit of such a straightforward compression algorithm is that fast image reconstruction at lower (spatial) resolution levels can be supported.

Assessment of compression/decompression quality is generally carried out using global measures such as mean square error (MSE) or peak signal-to-noise ratio (PSNR). For definitions of the latter, see section 5 below. Comparative assessment results can be found in UCLA (1997).

In this article the type of image which is of interest is one containing smooth features or point-like features. The discussion in this paper, and to some degree the methods used, are all based on such images. Images containing pronounced edges, which could perhaps be more typical of industrial vision, are not of direct interest to us in this context. See Xiong & Ramchandran (2000) for some discussion of wavelet-based approaches for addressing the compression of images where edges are important.

2 COMPRESSION SOFTWARE IN ASTRONOMY

In Louys, Starck, Mei, Bonnarel & Murtagh (1999a, see also Starck, Murtagh & Bijaoui, 1998) we compared a range of powerful compression methods – fractal, wavelet, pyramidal median, JPEG – with compression tools dedicated to astronomy such as Hcompress, FITSpress and mathematical morphology, and applied these to astronomical images.

2.1 Compression packages

Methods used in astronomy include Hcompress (White, Postman & Lattanzi, 1992), FITSpress (Press, 1992), and JPEG (Furht, 1995). These are all based on linear transforms, which in principle help to reduce the redundancy of pixel values in a block and decorrelate spatial frequencies or scales. Two other important methods have also been proposed for astronomical image compression: one using mathematical morphology, and another based on the pyramidal median transform (a nonlinear transform). These will be looked at below. A specific decompression postprocessing method has also been developed in Bijaoui, Bobichon & Huang (1996) in order to reduce artifacts relative to the Hcompress method. From the mainstream signal processing domain, two other recent approaches are worthy of attention. The first is based on fractals, and the second uses a bi-orthogonal wavelet transform. We first briefly review all of these methods, and then compare them in the framework of astronomical images.

Hcompress: Hcompress (White et al., 1992) was developed at Space Telescope Science Institute (STScI, Baltimore), and is commonly used to distribute archive images from the Digital Sky Survey DSS1 and DSS2. It is based on the Haar wavelet transform. The algorithm consists of

1. applying a Haar wavelet transform to the data,
2. quantizing the wavelet coefficients linearly as integer values,
3. applying a quadtree to the quantized value, and
4. using a Huffman coder.

Sources are available at http://www.stsci.edu/software/hcompress.html.

Hcompress with iterative decompression: Iterative decompression was proposed in Bijaoui et al. (1996) to decompress files which were compressed using Hcompress. The idea is to consider the decompression problem as a restoration problem, and to add constraints on the solution in order to reduce the artifacts.

FITSpress: FITSpress (Press, 1992) uses a threshold on very bright pixels and applies a linear wavelet transform using the Daubechies-4 filters. The wavelet coefficients are thresholded according to a noise threshold, quantized linearly and runlength encoded. This was developed at the Center for Astrophysics, Harvard. Sources (fitspress08.tar) are available at a number of locations on the web.

JPEG: JPEG is the standard image compression package for single frame images (Furht, 1995). It decorrelates pixel coefficients within 8×8 pixel blocks using the discrete cosine transform (DCT) and uniform quantization.

Wavelet transform: Various wavelet packages exist which support image compression, leading to more sophisticated compression methods. The wavelet

transform we used is based on a bi-orthogonal wavelet transform (using Antonini-Daubechies 9/7 coefficients) followed by entropy coding.

Fractal transform: The image is decomposed into blocks, and each block is represented by a fractal. See Fisher (1994) for more explanation.

Mathematical morphology: This method (Starck et al., 1998), denoted MathMorph in this paper, is based on mathematical morphology (erosion and dilation). It consists of detecting structures above a given level, the level being equal to the background plus three times the noise standard deviation. Then, all structures are compressed by using erosion and dilation, followed by quadtree and Huffman coding. This method relies on a first step of object detection, and leads to high compression ratios if the image does not contain a lot of what we might characterize as "continuous" (or smoothly varying) information, as is often the case in astronomy.

Pyramidal median transform: The principle of this compression method (Starck, Murtagh, Pirenne & Albrecht, 1996; Starck et al., 1998) denoted PMT here, is to select the information we want to keep, by using the pyramidal median transform, and to code this information without any loss. Thus the first phase searches for the minimum set of quantized multiresolution coefficients which produce an image of "high quality". The quality is evidently subjective, and we will define by this term an image with no visual artifact in the decompressed image; and the residual (original image – decompressed image) does not contain any evident structure.

Lost information cannot be recovered, so if we do not accept any loss, we have to compress what we take as noise too, and the compression ratio will be low (3 or 4 only).

The Pyramidal Median Transform (PMT) is obtained by the following algorithm:

1. Let $c_j = f$ with $j = 1$. f is the original image.
2. Determine $c^*_{j+1} = med(c_j, 2s+1)$ with $s = 1$. Here $med(c_j, 2s+1)$ is the convolution of image c_j with a median kernel of square dimensions $2s+1$.
3. The pyramidal multiresolution coefficients w_{j+1} are defined as: $w_{j+1} = c_j - c^*_{j+1}$.
4. Let $c_{j+1} = dec(c^*_{j+1})$ where the decimation operation, dec, entails 1 pixel replacing each 2×2 subimage.
5. Let $j \longleftarrow j+1$. Return to Step 2 so long as $j < S$.

Here the kernel remains the same during the iterations. The image itself, to which this kernel is applied, becomes smaller.

While this algorithm is computationally efficient, for ability to exactly reconstruct the input data we must use the following algorithm based on B-spline interpolation:

1. Take the lowest scale image, c_j.
2. Interpolate c_j to determine the next resolution image (of twice the dimensionality in x and y). Call the interpolated image c'_j.

3. Calculate $c_{j-1} \longleftarrow c'_j + w_j$.

4. Set $j \longleftarrow j - 1$. Go to Step 2 if $j > 0$.

This reconstruction procedure takes account of the pyramidal sequence of images containing the multiresolution transform coefficients, w_j. It presupposes, though, that a good reconstruction is possible. We ensure that by use of the following refined version of the Pyramidal Median Transform: using iteration, the coefficients, $w_{j+1} = c_j - c_{j+1}$, are improved relative to their potential for reconstructing the input image.

2.2 Remarks on these methods

The pyramidal median transform (PMT) is similar to the mathematical morphology (MathMorph) method in the sense that both try to understand what is represented in the image, and to compress only what is considered as significant. PMT uses a multiresolution approach, which allows more powerful separation of signal and noise. The latter two methods are both implemented in the MR software environment (MR1, 1999).

Each of these methods belongs to a general scheme where the following steps can be distinguished:

1. Decorrelation of pixel values inside a block, between wavelength, scales or shape, using orthogonal or nonlinear transforms.

2. Selection and quantization of relevant coefficients.

3. Coding improvement: geometrical redundancy reduction of the coefficients, using the fact that pixels are contiguous in an array.

4. Reducing the statistical redundancy of the code.

How each method realizes these different steps is indicated in Table 1.

Clearly these methods combine many strategies to reduce geometrical and statistical redundancy. The best results are obtained if appropriate selection of relevant information has been performed before applying these schemes.

For astronomical images, bright or extended objects are sought, as well as faint structures, all showing good spatial correlation of pixel values and within a wide range of graylevels. Noise background, on the contrary, shows no spatial correlation and fewer graylevels. The removal of noisy background helps in regard to data compression of course. This can be done with filtering, graylevel thresholding, or coarse quantization of background pixels. This is used by FITSpress, PMT and MathMorph which divide information into a noise part, estimated as a Gaussian process, and a highly correlated signal part. MathMorph simply thresholds the background noise estimated by a 3-sigma clipping, and quantizes the signal as a multiple of sigma (Huang & Bijaoui, 1991). FITSpress thresholds background pixels and allows for coarse background reconstruction, but also keeps the highest pixel values in a separate list. PMT uses a multiscale noise filtering and selection approach based on noise standard deviation estimation. JPEG and Hcompress do not carry out noise separation before the transform stage.

Table 1: Description and comparison of the different steps in the compression packages tested.

Software	Transform	Coefficient quantiz.	Coefficient organisation	Geometrical redundancy reduction	Statistical redundancy reduction
JPEG	DCT 8×8 pixels	Linear	Zigzag sequence	Runlength coding	Huffman
Hcompress	Haar 2×2 pixels	Linear	Pyramidal	Quadtree on bitplanes	Huffman
FITSpress	Wavelets Daub-4	Linear	Increasing Resolution	Runlength coding	Huffman
MR/1 PMT	Pyramidal Med. Trans.	Linear / Noise est.	Decreasing Resolution	Quadtree on bitplanes	Huffman
MR/1 MathMorph	Erosion/ Dilation	Linear / Noise est.	–	Quadtree on bitplanes	Huffman

2.3 Identifying the information loss

Apart from signal-to-noise discrimination, information losses may appear after the transforms at two steps: coefficient selection and coefficient quantization. The interpretable resolution of the decompressed images clearly depends upon these two steps.

If the spectral bandwidth is limited, then the more it is shortened, the better the compression rate. The coefficients generally associated with the high spatial frequencies related to small structures (point objects) may be suppressed and lost. Quantization also introduces information loss, but can be optimized using a Lloyd-Max quantizer for example (Proakis, 1995).

All other steps, shown in Table 1, such as reorganizing the quantized coefficients, hierarchical and statistical redundancy coding, and so on, will not compromise data integrity. This statement can be made for all packages. The main improvement clearly comes from an appropriate noise/signal discrimination and the choice of a transform appropriate to the objects' signal properties.

3 EVALUATION

In identifying the needs for compression, different strategies can be used.

1. Compression without visual loss. This means that one cannot see the difference between the original image and the decompressed one. Generally, compression ratios between 10 and 20 can be obtained.

2. Good quality compression: the decompressed image does not contain any artifact, but some information is lost. Compression ratios up to 40 can be obtained in this case.

3. Fixed compression ratio: for some technical reasons, one may decide to compress all images with a compression ratio higher than a given value, whatever the effect on the decompressed image quality.

Table 2: List of criteria for comparison of compression methods for various types of astronomical image-based application. Recal. = recalibration.

Application type / **Comparison criterion**	**Quick view**	**Catalog overlay: cross-** corr.	**Source ext. and Cross-id.**	**Deep detection**	**Recal.**
quality: visual	medium	high	medium	indifferent	high
quality: precision	low	medium	high	very high	high
Transfer + comp. speed	very fast	fast	medium	slow	medium
Progressive vision	yes	yes	no	no	no

4. Signal-to-noise separation: if noise is present in the data, noise modeling can allow very high compression ratios just by including some type of filtering in the transform space during the compression.

Following the image types, and the selected strategy, the optimal compression method may vary.

3.1 Quality assessment

In order to compare the different compression methods, various characteristics are of importance (see Table 2).

Progressive vision is useful in the context of quick views (for example on the web) and image overlays, where the user can decide when the quality of a displayed image is sufficient. However this feature is not required for more quantitative tasks.

The requirement of speed of display (transfer time plus processing time) is usually critical for applications related to progressive vision.

The estimation of the quality of a compression method and rate compared to others is based on the quality of restitution of the relevant information, which is always relative to the type of application. For good quality quick views of a given area, label and image database overlays, and cross-correlation of features at different wavelengths, the required quality will be essentially qualitative: good geometry of the objects, no visual artifacts, good contrast, etc.

For cross-identification processes, and any situation where recalibration to improve astrometry and photometry is needed, or reprocessing of object detection where some were obviously missed, or false merging of distinct objects – for such cases, quality estimation must be a quantitative process. The loss of information can be measured by the evolution of "relevant parameters" varying according to compression rate and method.

Quality criteria which can be used for estimating the merits and performances of a compression method fall under these headings: Visual aspect, signal-to-noise ratio, detection of real and faint objects or features, object morphology, astrometry, and photometry.

In our studies, quality was quantified from visual fidelity, and from photometric (i.e., integrated intensity) and astrometric (i.e., positional) measurements. Computational requirements of each method were noted. We also reviewed the implications of web-based storage and transmission, stressing the importance of

progressive vision. In Louys, Starck & Murtagh (1999b) we summarized lossless compression algorithms. Figures 1 and 2 illustrate these results. Figure 3 shows the results of astrometry (positional information) testing. Tables 1 and 2 summarize these results.

Lossless compression is based on the Haar wavelet transform, implemented with the lifting scheme, and followed by entropy coding. Lower resolution versions of the image data, in the context of this transform, correspond to average values in larger image areas. Photometric properties of a lower resolution image are therefore exact averages of the photometric measurements made at finer resolution levels.

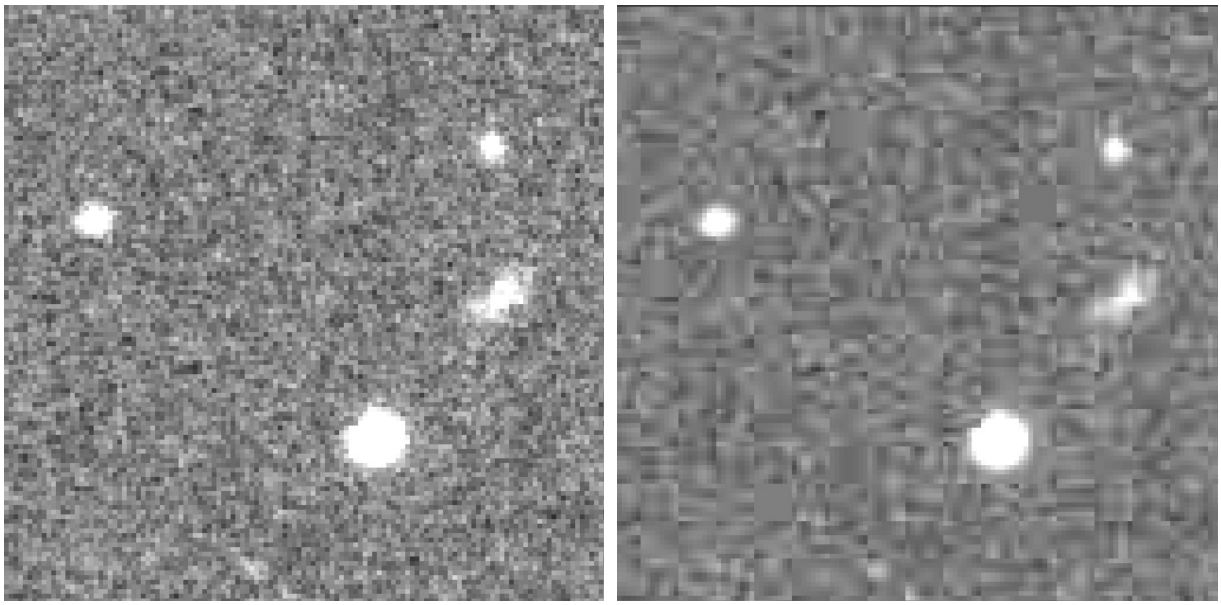

Figure 1: Left: Original image, subimage extracted from 1024×1024 patch, extracted in turn from the central region of ESO7992v, a digitized Schmidt photographic plate. Right: JPEG compressed image at 40:1 compression rate.

An important aspect of lossy compression in the case of medical and scientific images is that noise be accurately defined and removed. Extensive modeling of noise for use in image compression, filtering, deconvolution and other operations, is supported in the MR software environment (MR1, 1999; see also Starck et al. 1998).

We have comprehensively examined compression performance on large numbers of astronomy images. Consider for example a 12451 × 8268 image from the CFH12K detector at the CFHT (Canada-France-Hawaii Telescope), Hawaii. See Figure 4. A single image is 412 MB. As astronomy detectors tend towards 16000 × 16000 in image dimensions – the case of the UK's Vista telescope now being designed for operation in Chile for instance – it is clear that compression and delivery technologies are very much needed. A typical observing night gives rise to terabytes of data, and image repositories are measured in petabytes.

Using denoising (visually lossless) compression, we compressed the CFH12K

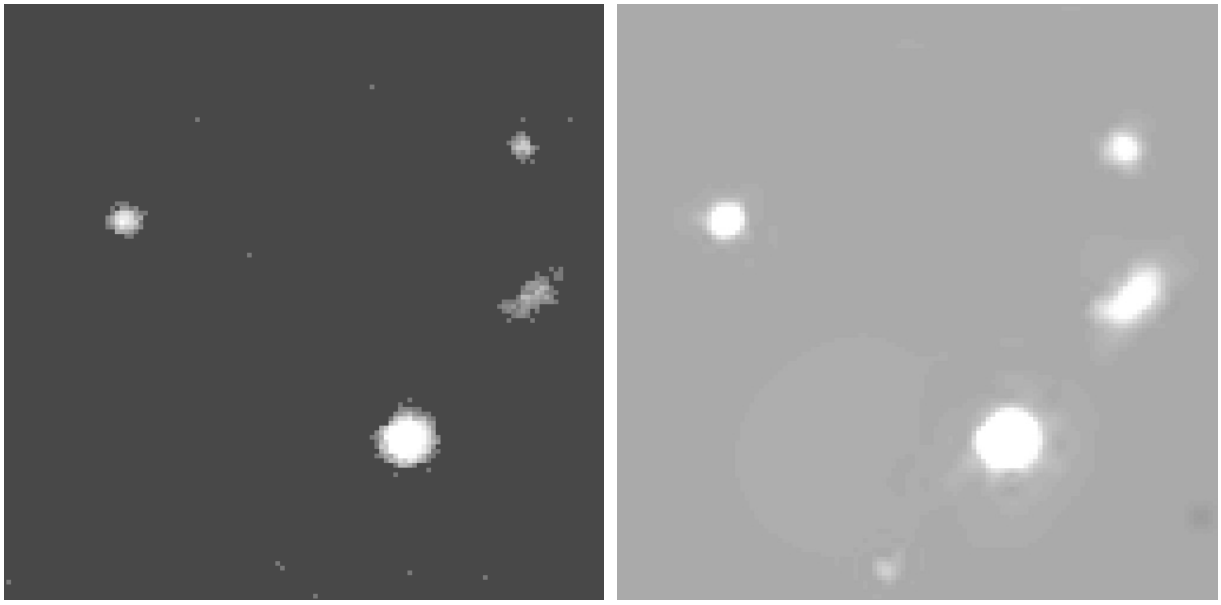

Figure 2: Left: MathMorph-compressed image of the same patch, at 203:1 compression rate. Right: PMT-compressed image at 260:1 compression rate.

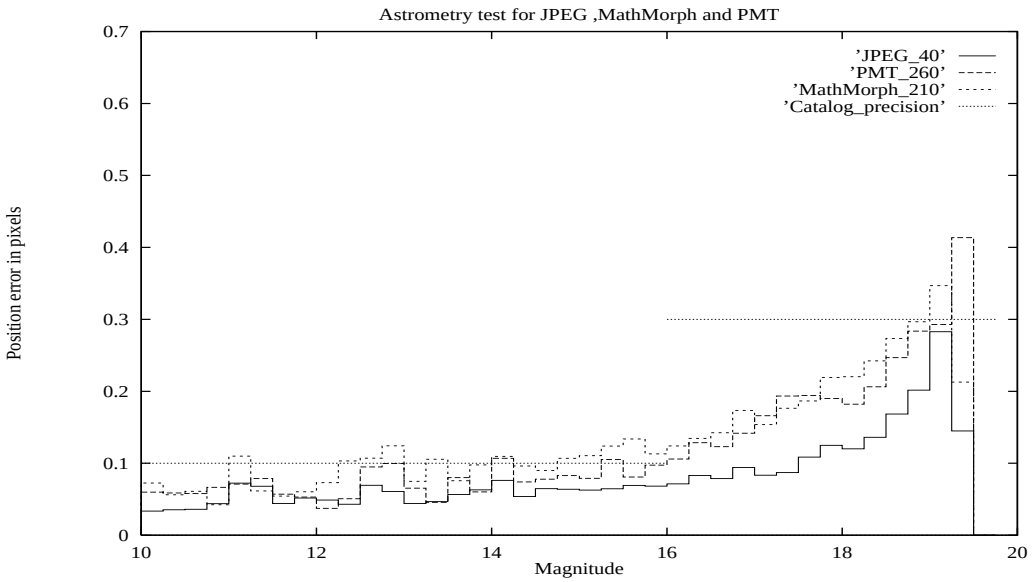

Figure 3: Mean error in astrometry, by interval of 0.25 magnitude, for images compressed 40 times by JPEG, 260 times by PMT, and 210 times for MathMorph.

Table 3: Compression of a 1024 × 1024 16-bit integer image. Platform: Sun Ultra-Enterprise, 250 MHz. Noted are whether artifacts are created, typical compression ratios obtained, and whether progressive transmission implementations are currently available.

	Compression time (secs.)	Decompression time (secs.)	Artifact	Compression 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
Mathematical Morphology	13	7.86	N	< 210	N
Hcompress	3.29	2.82	Y	270	Y (in C)
Hcompress + iterative reconstruction	3.29	77	N	270	N
Pyramidal Median Trans.	7.8	3.1	N	270	Y (in Java)

image to 4.1 MB, i.e. less than 1% of its original size. Compression took 789 seconds on an Ultra-Sparc 10. Decompression to the fifth resolution scale (i.e., dimensions divided by 2^5) took 0.43 seconds. For rigorously lossless compression, compression to 97.8 MB, i.e. 23.75% of the original size, took 224 seconds, and decompression to full resolution took 214 seconds. Decompression to full resolution by block was near real-time.

4 DECOMPRESSION BY SCALE AND BY REGION

Support of the transfer of very large images in a networked (client-server) setting requires compression and prior noise separation. In addition, progressive transfer may be supported, or delivery by scale and by image region. For such additional functionality, wavelet transform based methods are very attractive because they integrate a multiresolution concept in a natural way. Our MR software system (MR1, 1999) contains a "LIVE: Large Image Visualization Environment" prototype, which is Java-based at the client end, and allows access to differing resolution levels as well as block-sized regions of the compressed image data. Figure 5 exemplifies the overall design of a system allowing for decompression by resolution scale and by region block.

Table 4: Compression of a 1024 × 1024 16-bit integer image. Platform: 2-processor Sun Ultra-Sparc 250 MHz.

Software	Compression time (secs.)	Decompression time (secs.)	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

Systems have been prototyped which allow for decompression at full resolution in a particular block, or at given resolutions in regions around where the user points to with a cursor. The former is shown in Figure 6. Rigorously lossless compression is obtained, to 4.9% of original image size. Decompression is carried out in effective real time of a region of interest surrounding the tibia fracture site. A description of this work can be found in Farid, Murtagh, Louys & Starck (2001), and the on-line demonstrator is available at `http://strule.cs.qub.ac.uk/imed.html`

5 APPLICATION TO MEDICAL IMAGE COMPRESSION

Table 5: Compression results on image shown in Figure 6. Accumulated user times in seconds on 350 MHz Sparc workstation. Input image size (at 16 bits per pixel): 7,535,520 bytes. bpp = bits per pixel, with 16 as the value pertaining to the input image. MSE = mean square error. PSNR = peak signal to noise ratio (see text for definitions of 1 and 2).

	Bi-orthogonal 9/7	Pyramidal median	Lossless Haar / lifting
compression time	11 secs.	10 secs.	5 secs.
decompression time	15 secs.	6secs.	6 secs.
size (bytes)	125,225	311,792	732,266
compression ratio	1.66%	4.14%	9.72%
compression bpp	0.266	0.662	1.555
MSE	0.78	0.76	–
PSNR (1)	97.40 dB	97.52 dB	–
PSNR (2)	193.73 dB	193.85 dB	–

Figure 6 shows a fractured tibia with an intramedullary nail fitted, to aid in recovery. This image is originally of dimensions 2040 × 1760 in DICOM 16-bit unsigned integer format. Results are quoted relative to this.

A bi-orthogonal Antonini-Daubechies wavelet transform with 9/7 tap filters was

Figure 4: CFHT image of dimensions 12451 by 8268 used for compression timing experiments.

used for lossy compression, and as an alternative the pyramidal median transform. Lossless compression is based on the lifting scheme which can guarantee that integer values are used at all times, and the Haar wavelet transform.

Visual results of uncompressed images following lossy compression are excellent. Table 5 presents quantitative results. In all cases, compression was carried out blockwise, with block dimension 128. Definition 1 of PSNR is $10\log_{10}(65536^2/\text{MSE})$ and definition 2 of PSNR is $20\log_{10}(65536^2/\text{RMS})$, where MSE and RMS are mean square error, and root mean square error, respectively. The image used has a lot of "background", which explains why the results are so dramatically good – indeed, even for lossless compression.

Digital, or digitized, mammograms can be very large. As an example, digitizing a mammogram film at 42 micron steps can yield an image of dimensions 4240×5670, and at 16 bits per pixel, that implies a file size of 48 MB.

Figure 7 shows a 512×512 subimage extracted from a mammogram. Lossless

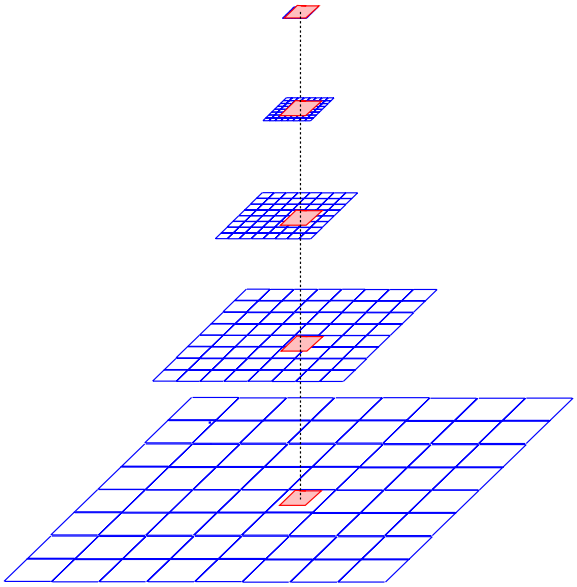

Figure 5: Example of large image, compressed by block, and represented at five resolution levels. At each resolution level, the visualization window is superimposed at a given position. At low resolution, the window covers the whole image, while at the full resolution level, it covers only one block.

compression, again based on the Haar wavelet transform implemented with the lifting scheme, provided for compression to 17% of the original image size, and was near real-time (i.e., rounded, 0 seconds). Decompression was also near real-time. The decompressed image was rigorously identical to the input image. In bits per pixel, this lossless compression performance was 0.34 bpp.

Lossy compression which improved on these compression rates by a factor of two required 1 second to carry out.

In the case considered, rigorously lossless compression has provided excellent results. More typical lossless compression rates are as shown in Table 4. In this case, compression strategies such as the following may be of use. First, provide quick view functionality, based on lossy compressed images. Secondly, provide access to the "completion" of the image, by adding in the difference map between original data and lossy compressed data. Such an approach is premised on the fact that some, and maybe considerable, assessment and diagnosis can be carried out on the visually lossless image data. On demand, the user has access to the full data.

6 CONCLUSION

In this article we have presented outstanding compression and decompression results on very large scientific images. We have described demonstrator systems which we have implemented for web-based real-time decompression and the delivery of image regions of interest.

Wavelets and multiscale transforms offer many advantageous properties: a framework for excellent quality when coding or encoding information; easy incorpora-

tion of noise models which are crucial for lossy, visually acceptable compression; controlled and scalable implementation and operation; and targeted dissemination, delivery and display.

This algorithmic framework is well-adapted for the support of access to, and scientific analysis of, large image repositories.

7 ACKNOWLEDGEMENTS

The CFHT image was provided by Y. Mellier. Others whom we wish to acknowledge are J. Guibert, students Dubaj, Couvidat and Skalkovski, Li H. and A. Bijaoui for helping in tuning the MathMorph approach, and O. Bienaymé. The tibial fracture image was from D. Marsh and C. McGivern, and the mammogram from R.J. Hanisch.

8 REFERENCES

Bijaoui, A., Bobichon, Y. & Huang, L. (1996) Digital image compression in astronomy: morphology or wavelets. *Vistas in Astronomy 40*, 587–594.

Farid, M., Murtagh, F., Louys, M. & Starck, J.L. (2001) Compression and delivery for the support of distributed image databases. In Winstanley, A.C. (Ed.), *Proc. IMVIP – Irish Machine Vision and Image Processing Conference*, Maynooth: NUI Maynooth, pp. 249–252.

Fisher, F. (1994) *Fractal Image Compression: Theory and Applications*, Berlin: Springer-Verlag.

Furht, B. (1995) A survey of multimedia compression techniques and standards. Part II: Video compression. *Real-Time Imaging 1*, 49–67.

Huang, L. & Bijaoui, A. (1991) Astronomical image data compression by morphology skeleton transformation. *Experimental Astronomy 1*, 311–327.

Louys, M., Starck, J.L., Mei, S., Bonnarel, F. & Murtagh, F. (1999a) Astronomical image compression. *Astronomy and Astrophysics Supplement 136*, 579–590.

Louys, M., Starck, J.L. & Murtagh, F. (1999b) Lossless compression of astronomical images. *Irish Astronomical Journal 26*, 119–122.

MR/1, 2, 3 (1999) *Multiresolution Image and Data Analysis Software Environment*, Multi Resolutions Ltd., http://www.multiresolution.com

Press, W.H. (1992) Wavelet-based compression software for FITS images. In Worrall, D.M., Biemesderfer, C. & Barnes, J. (Eds.), *Astronomical Data Analysis Software and Systems I*, San Francisco: Astronomical Society of the Pacific, pp. 3–16.

Proakis, J.G. (1995) *Digital Communications*, New York: McGraw-Hill.

Starck, J.L., Murtagh, F. & Bijaoui, A. (1998) *Image Processing and Data Analysis: The Multiscale Approach*, Cambridge: Cambridge University Press.

Starck, J.L., Murtagh, F., Pirenne, B. & Albrecht, M. (1996) Astronomical image compression based on noise suppression. *Publications of the Astronomical Society of the Pacific 108*, 446–455.

UCLA (1997) Image Communications Laboratory, University of California Los Angeles, Wavelet image coding: PSNR results. www.icsl.ucla.edu/∼ipl/psnr_results.html.

Welstead, S. (1999) *Fractal and Wavelet Image Compression Techniques*, Bellingham: SPIE Press.

White, R., Postman, M. & Lattanzi, M. (1992) Compression of the Guide Star Digitised Schmidt plates. In MacGillivray, H.T. & Thompson, E.B. (Eds.), *Digitized Optical Sky Surveys*, Dordrecht: Kluwer, pp. 167–175.

Xiang Xiong & Ramchandran K. (2000) Wavelet image compression. In Bovik, A. (Ed.) *Handbook of Image and Video Processing*, New York: Academic, pp. 495–512.

Figure 6: X-ray image of patient leg with intramedullary nail.

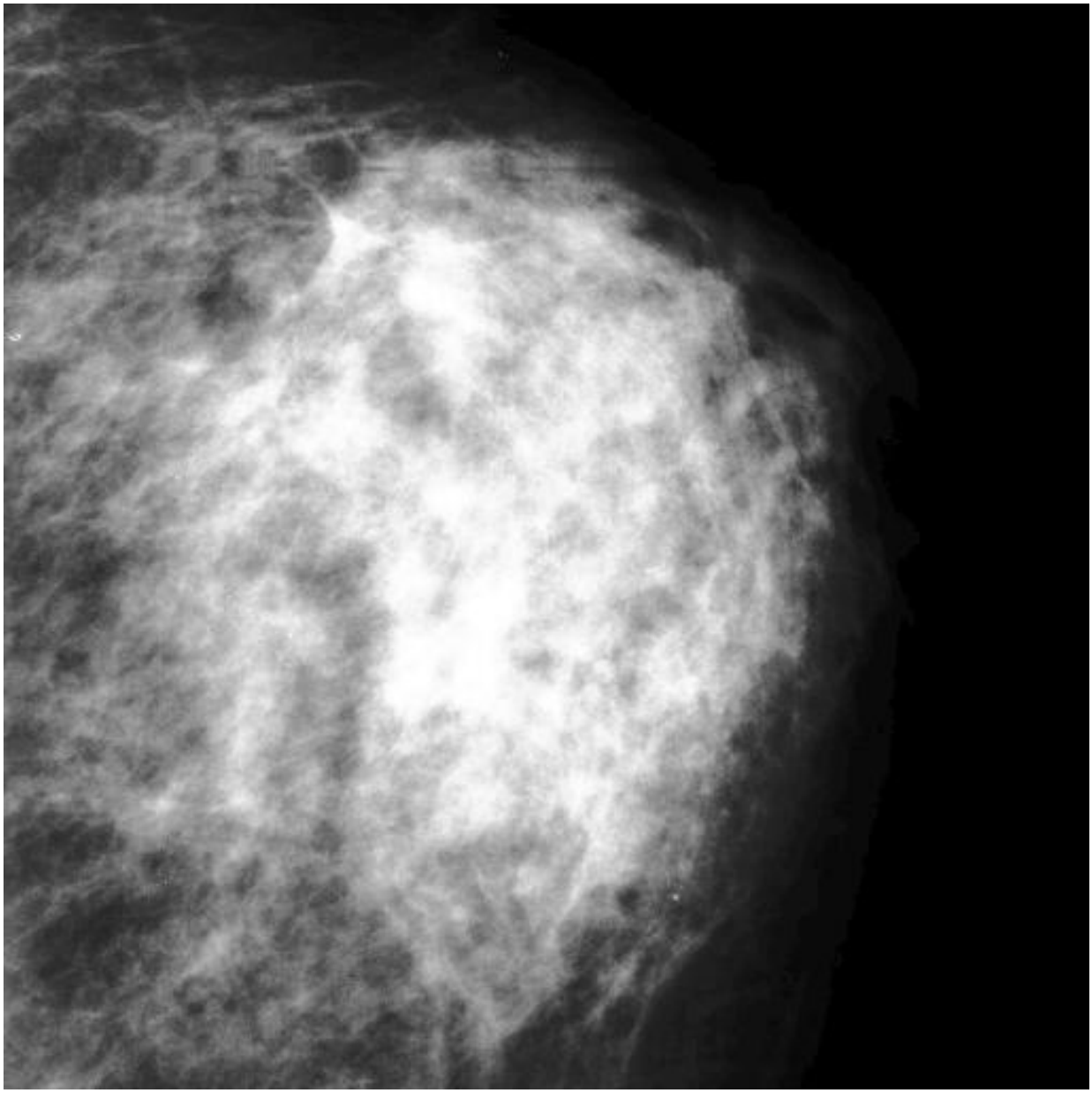

Figure 7: Part of a digitized mammogram, shown histogram-equalized.