On the Selection of Image Compression Algorithms

Chaur-Chin Chen *
Department of Computer Science
National Tsing Hua University
Hsinchu 300, Taiwan
e-mail: cchen@cs.nthu.edu.tw

Abstract

This paper attempts to give a recipe for selecting one of the popular image compression algorithms based on (a) Wavelet, (b) JPEG/DCT, (c) VQ, and (d) Fractal approaches. We review and discuss the advantages and disadvantages of these algorithms for compressing gray-scale images, give an experimental comparison on four 256×256 commonly used images, Jet, Lenna, Mandrill, Peppers, and one 400×400 fingerprint image. Our experiments show that all of the four approaches perform satisfactorily when the 0.5 bits per pixel (bpp) is desired. However, for a low bit rate compression like 0.25 bpp or lower, the embedded zerotree wavelet (EZW) approach and DCT-based JPEG approach are more practical.

1 Introduction

As media communication grows and video on demand is desired, image data compression has received an increasing interest. The purpose of image compression is to achieve a very low bit rate representation, for example, MPEG-4 [21] aims at 64K bits per second while preserving a high visual quality of decompressed images. The fundamental techniques of video compression are based on the schemes of still gray level image compression.

This paper reviews and lists the characteristics of four popular image compression algorithms based on (a) Wavelet, (b) JPEG/DCT, (c) VQ, and (d) Fractal methods, gives experimental comparisons of these algorithms on several real images. The purpose is to give a recipe for selecting an appropriate image compression algorithm for the problems in hand. The PSNR (peak singal-to-noise ratio) value used to measure the differ-

ence between a decoded image $\hat{f}$ and its original image f is defined as follows. In general, the larger PSNR value, the better decoded image quality.

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [\hat{f}(i,j) - f(i,j)]^2 \quad (1)$$

$$PSNR = 10 \log_{10} \left[\frac{255 \times 255}{MSE} \right] \; dB \quad (2)$$

2 Review of Compression Algorithms

The goal of image compression is to save storage space and to reduce transmission time for image data. It aims at achieving a high compression ratio (CR) while preserving good fidelity of decoded images. The techniques used to compress/decompress a single gray level image are expected to be easily modified to encode/decode color images and image sequences. Recent compression methods can be briefly classified into four categories: (a) Wavelet, (b) JPEG/DCT, (c) VQ, and (d) Fractal methods, which are briefly reviewed below.

2.1 Wavelet Compression

Image compression based on wavelet transforms has recently received an increasing interest [2, 11, 16, 17]. The current state-of-the-art wavelet approach applies a wavelet transform on images in a pyramid fashion up to the desired scale using the theory of multiresolution signal decomposition with the wavelet representation [6, 13] and the concept of embedded zerotree wavelet (EZW) based on the *decaying spectrum* hypothesis [17]. In a pyramidal structure after a certain scale of wavelet transforms on an image, an algorithm [17] successively determines if a coefficient is significant in the spatial-frequency domain to form a significance map consisting of the sign (+ or −) of a significant coefficient, an insignificant symbol, and a zerotree symbol.

It assumes that wavelet coefficients of an image in the finer resolutions corresponding to a zerotree mark have smaller magnitudes than the one marked as zerotree in a coarser resolution for this image according to a practical, but false *decaying spectrum* hypothesis. An algorithm has been widely tested and shown to be very effective [16, 16].

2.2 JPEG Compression

The JPEG/DCT still image compression has become a standard recently [14, 18]. To exploit this method, an image is first partitioned into nonoverlapped 8×8 blocks. A discrete Cosine transform (DCT) [1, 14] is applied to each block to convert the gray levels of pixels in the spatial domain into coefficients in the frequency domain. The coefficients are normalized by different scales according to the quantization table provided by the JPEG standard conducted by some psychovisual evidence. The quantized coefficients are rearranged in a zigzag scan order to be further compressed by an efficient lossless coding strategy such as runlength coding, arithmetic coding, or Huffman coding [14]. The decoding is simply the inverse process of encoding. So, the JPEG compression takes about the same time for both encoding and decoding. The encoding/decoding algorithms provided by an independent JPEG group [20] are available for testing real-world images.

The information loss occurs only in the process of coefficient quantization. The JPEG standard defines a standard 8×8 quantization table [14] for all images which may not be appropriate. To achieve a better decoding quality of various images with the same compression by using the DCT approach, an adaptive quantization table may be used instead of using the standard quantization table.

2.3 VQ Compression

The fundamental idea of VQ for image compression is to establish a codebook consisting of codevectors such that each codevector can represent a group of image blocks of size $m \times m$, ($m=4$ is always used). An image or a set of images is first partitioned into $m \times m$ nonoverlapping blocks which are represented as m^2-tuple vectors, called training vectors. The size of training vectors can be very large. For example, a 512×512 image contributes 16,384 training vectors. The goal of codebook design is to establish a few representative vectors, called codevectors of size 256 or 512, from a set of training vectors. The encoding procedure is to look for a closest codevector in the codebook for each nonoverlapped 4×4 block of an image to be encoded. The most important work is to design a versatile codebook. Nasrabadi and King [15] give a good review of VQ. Chen's comparison [4] indicates that a codebook developed based on LBG [12] algorithm generally has higher PSNR values over some other schemes despite its slow off-line training. In this paper, we adopt LBG [12] algorithm for training a codebook of size 256×256 to meet a desired 0.5 bpp compression ratio.

2.4 Fractal Compression

Fractal image coding was introduced in the late 1980s and early 1990s [10, 22]. It is used for encoding/decoding images in Encarta/Encyclopedia [3]. Fractal coding is based on the Collage theorem and the fixed point theorem [3] for a local iterated function system consisting of a set of contraction affine transformations [3, 9]. A fractal compression algorithm first partitions an image into nonoverlapping 8×8 blocks, called range blocks and forms a domain pool containing all of possibly overlapped 16×16 blocks, associated with 8 isometries from reflections and rotations, called domain blocks [8, 10]. For each range block, it exhaustively searches, in a domain pool, for a best matched domain block with the minimum square error after a contractive affine transform is applied to the domain block. A fractal compressed code for a range block consists of quantized contractivity coefficients in the affine transform, an offset which is the mean of pixel gray levels in the range block, the position of the best matched domain block and its type of isometry. The decoding is to find the fixed point, the decoded image, by starting with any initial image. The procedure applies a *compressed* local affine transform on the domain block corresponding to the position of a range block until all of the decoded range blocks are obtained. The procedure is repeated iteratively until it converges (usually in no more than 8 iterations). Two serious problems that occur in fractal encoding are the computational demands and the existence problem of best range-domain matches [9]. The most attractive property is the resolution-independent decoding property. One can enlarge an image by decoding an encoded image of smaller size so that the compression ratio may increase exponentially [3, 8]. An algorithm based on [10] using range and domain block matches of fixed sizes is written and is used for a comparison in this paper [5]. Other algorithms using various block sizes of domain and range blocks associated with a quadtree structure can be found in [9].

Method	Advantages	Disadvantages
Wavelet	• high compression ratio ratio • state-of-the-art	• coefficient quantization • bit allocation
JPEG (DCT)	• current standard	• coefficient quantization • bit allocation
VQ	• simple decoder • no coefficient quantization	• slow codebook generation • small bpp
Fractal	• good mathematical encoding frame • resolution-free decoding	• slow encoding

Method	Compression ratio	Appeared in
Wavelet	$\gg 32$	1992 [2] 1993 [17] 1996 [16]
JPEG (DCT)	≤ 50	1974 [1] 1993 [14]
VQ	< 32	1980 [12] 1989 [7]
Fractal	≥ 16	1992 [10] 1992 [8]

Table 1: Characteristics of four popular image coding methods

2.5 Summary

We have briefly discussed four popular image coding algorithms. Major characteristics of these approaches based on (a) Wavelet, (b) JPEG/DCT, (c) VQ, and (d) Fractal methods are briefly summarized in Table 1. An experimental comparison is given in the next section. A recipe of how to select a compression algorithm for the problem at hand is given in Section 4.

3 Experimental Comparison

Image compression algorithms based on EZW [16], JPEG/DCT [20], VQ [4], and Fractal [5] methods were tested for four 256×256 real images: Jet, Lenna, Mandrill, Peppers, and one 400×400 fingerprint image. The original images of *Lenna* and *fingerprint* are shown in Figure 1. The results of performance on a Sun Ultra 1 Sparc workstation running Solaris OS 5.5.1 are reported in Tables 2 - 3 and Figures 2 - 3. The decoded images based on the four approaches are shown in Figures 1 and 2. The associated PSNR values and encoding/decoding times shown in Tables 2 - 3 for the

Algorithm	PSNR values (in dB)			
	Jet	Lenna	Mandrill	Peppers
Wavelet	32.48	34.66	26.54	34.99
JPEG	30.39	31.73	25.15	31.95
VQ	26.76	29.28	24.45	29.12
Fractal	26.70	29.04	24.29	29.13

	CPU time	
	Encoding	Decoding
Wavelet	0.35 sec	0.27 sec
JPEG	0.12 sec	0.12 sec
VQ	2.45 sec	0.18 sec
Fractal	5.65 hrs	1.35 sec

Table 2: Performance of coding algorithms on various 256×256 images.

test images indicate that all the four approaches are satisfactory at 0.5 bpp request (CR=16). However, the EZW [16, 17] has significantly larger PSNR values and a better visual quality of decoded images compared with the other approaches.

At a desired compression of 0.25 bpp (CR=32) for the fingerprint image, the commonly used VQ can not be tested, and the fractal coding can not be achieved unless resolution-free decoding property is utilized which is not useful for the current purpose; both EZW [16] and JPEG [20] approaches perform well, and the results of EZW have significant larger PSNR values than that of JPEG.

For an image with highly textured regions such as *Mandrill*, the PSNR values of the four methods are significantly lower than those of images with a large portion of smooth regions such as *Lenna* and *Peppers*. Image *Jet* contains a large portion of sharp edges and textures, so its PSNR values of the four methods are between those of *Mandrill* and *Peppers*.

4 Conclusion

We have reviewed and summarized the characteristics of four up-to-date image coding algorithms based on Wavelet, JPEG/DCT, VQ, and Fractal approaches. Experimental comparisons on four 256×256 commonly used images, Jet, Lenna, Mandrill, Peppers, and one 400×400 fingerprint image suggest a recipe described as follows. Any of the four approaches is satisfactory when the 0.5 bits per pixel (bpp) is requested. However, for a very low bit rate, for example 0.25 bpp or lower, the embedded zerotree wavelet (EZW) approach

Algorithm	0.50 bpp		
	PSNR values	Encoding	Decoding
Wavelet	36.71	0.8 sec	0.7 sec
JPEG	34.27	0.2 sec	0.2 sec
VQ	28.26	6.0 sec	0.7 sec
Fractal	27.21	6.3 hrs	3.5 sec
Algorithm	0.25 bpp		
	PSNR value	Encoding	Decoding
Wavelet	32.47	0.7 sec	0.5 sec
JPEG	29.64	0.2 sec	0.2 sec
VQ	N/A	N/A	N/A
Fractal	N/A	N/A	N/A

Table 3: Performance of coding algorithms on a 400×400 fingerprint image.

(a) (b)

(c) (d)

Figure 2: Decoded Lennas by (a) Wavelet, (b) JPEG, (c) VQ, and (d) Fractal algorithms.

(a) (b)

Figure 1: Original images of (a) Lenna and (b) fingerprint.

(a) (b)

(c) (d)

Figure 3: Decoded fingerprints by (a) Wavelet, (b) JPEG, (c) VQ, (d) Fractal algorithms.

is superior to other approaches. For practical applications, we conclude that (1) Wavelet based compression algorithms [2, 11, 16, 17, 23] are strongly recommended, (2) DCT based approach might use an adaptive quantization table, (3) VQ approach is not appropriate for a low bit rate compression although it is simple, (4) Fractal approach should utilize its resolution-free decoding property for a low bit rate compression, (5) Hybrid compression algorithms based on some of these four approaches may be pursued to achieve a higher compression ratio while preserving a better quality of up-to-date decoded images.

References

[1] N. Ahmed, T. Natarajan, and K.R. Rao, Discrete cosine transform, *IEEE Trans. on Computers*, vol. 23, 90-93, 1974.

[2] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, Image coding using wavelet transform, *IEEE Trans. on Image Processing*, vol. 1, 205-220, 1992.

[3] M.F. Barnsley and L.P. Hurd, Fractal Image Compression, AK Peters, Ltd. Wellesley, Massachusetts, 1993.

[4] Y.W. Chen, Vector Quantization by principal component analysis, M.S. Thesis, National Tsing Hua University, June, 1998.

[5] H.S. Chu, A very fast fractal compression algorithm, M.S. Thesis, National Tsing Hua University, June, 1997.

[6] I. Daubechies, Ten Lectures on Wavelet Analysis. *SIAM*, 1992.

[7] W.H. Equitz, A new vector quantization clustering algorithm, *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 37, 1568-1575, 1989.

[8] Y. Fisher, Fractal Image Compression, SIG-GRAPH Course Notes, 1992.

[9] Y. Fisher, Editor, Fractal Image Compression: Theory and Applications, Springer-Verlag, 1994.

[10] A.E. Jacquin, Image coding based on a fractal theory of iterated contractive image transformations. *IEEE Trans. on Image Processing*, vol. 1, 18-30, 1992.

[11] A.S. Lewis and K. Knowles, Image compression using 2D wavelet transform, *IEEE Trans. on Image Processing*, vol. 1, 244-250, 1992.

[12] Y. Linde, A. Buzo, and R. M Gray, An algorithm for vector quantizer design, *IEEE Trans. on Communications*, vol. 36, 84-95, 1980.

[13] S.G. Mallat, A theory for multiresolution signal decomposition: the wavelet representation, *IEEE Trans. on PAMI*, vol. 11, 674-693, 1989.

[14] W.B. Pennebaker, J. Mitchell, JPEG Still Image Compression Standard, New York: Van Nostrand Reinhold, 1993.

[15] N.M. Nasrabadi and R.A. King, Image coding using vector quantization: a review, *IEEE Trans. on Communications*, vol. 36, 957-571, 1988.

[16] A. Said and W.A. Pearlman, A new, fast, and efficient image codec based on set partitioning in hierarchical trees, *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 6, 243-250, 1996.

[17] J.M. Shapiro, Embedded image coding using zerotree of wavelet coefficients, *IEEE Trans. on Signal Processing*, vol. 41, 3445-3462, 1993.

[18] G.K. Wallace, The JPEG still picture compression standard, *Communication ACM*, vol. 34, 31-44, 1991.

[19] C.J. Wein and I.F. Blake, On the performance of fractal compression with clustering, *IEEE Trans. on Image Processing*, vol. 5, 522-526, 1996.

[20] ftp.uu.net:/graphics/jpeg/jpegsrc.v6a.tar.gz

[21] http://wwwam.hhi.de/mpeg-video/#MPEG7

[22] http://links.uwaterloo.ca

[23] http://www.amara.com/current/Wavelet.html