2.

Fractal Image Compression Using Iterated Transforms: Applications to DTED

E.W. Jacobs and R.D. Boss
NCCOSC RDT&E Div
Code 573, San Diego, CA 92152-5000

ABSTRACT: A review of iterated transformation image compression is presented. Generalization of simple iterated function system fractal generating algorithms to an automated iterated transformation algorithm used to compress greyscale images is reviewed. Compressed images from the Digital Terrain Elevation Database are presented, and compared with encodings using adaptive discrete cosine transformations, and mean residual vector quantization image compression techniques.

1. INTRODUCTION

Because of the increasing use of digital imagery, there is currently considerable interest in the image compression problem. In particular, image compression is a current and growing necessity for Navy applications including storage and transmittal of maps, intelligence photographs, weather information, etc. General interest in image compression has led to the establishment by the Joint Photographic Experts Group of a standard based on discrete cosine transforms (ADCT). There is also an on going effort in the research community to design improved vector quantization (VQ) methods, and to develop methods which utilize wavelet transformations. A relatively new approach to the image compression problem, iterated transformations, has been presented by Jacquin [1,2]. This method has its foundation in the theory of iterated function systems (IFSs), developed by Hutchinson [3] and Barnsley [4], and recurrent iterated function systems [5]. The iterated transform algorithm has received particular interest because of the *fractal* nature of the encoded images, and because there has been much speculation, but little information available on the capabilities of the method. The first sections of this paper review the basic methodology of the iterated transform image compression technique. This is followed by a section on the compression of the Digital Terrain Elevation Database (DTED) in which results obtained using iterated transformations are compared to ADCT and VQ methods.

2. BACKGROUND: SIMPLE EXAMPLES

This example serves as a simple illustration of some concepts involved in the iterated transform image-encoding scheme. This example is based on iterated function systems. The main concept is that the image of a set (a Sierpinski gasket, in this case) can be reconstructed from a set of transformations which may take less memory to store than the original image.

Consider the three transformations shown in figure 1. They are

$$w_1\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

$$w_2\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{2} \end{bmatrix},$$

and

$$w_3\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \frac{1}{2} \\ 0 \end{bmatrix}.$$

For any set S, let

$$W(S) = \bigcup_{i=1}^{3} w_i(S).$$

Denote the n-fold composition of W with itself as $W^{\circ n}$. Define $A_n = W(A_{n-1}) = W^{\circ n}(A_0)$ and arbitrarily choose A_0 as the unit square with lower left corner at the origin (i.e., $A_0 = \{(x,y)|0 \leq x \leq 1, 0 \leq y \leq 1\}$). Then as $n \to \infty$, the set A_n converges to a limit set A_∞. In fact, for any compact set $S \subset R^2$, $W^{\circ n}(S) \to A_\infty$ as $n \to \infty$. Figure 2 shows $A_0, A_1, A_2, A_3, A_4, A_5$ and A_∞.

That all compact initial sets converge under iteration to A_∞ is important—it means that the set A_∞ is defined by the w_i only.

Each w_i is determined by 6 real values, so that for this example 18 floating point numbers are required. In single precision, this requires 72 bytes. The memory required to store an image of the set depends on the resolution; the

U.S. Government work not protected by U.S. copyright

Reproduced with permission of copyright owner. Further reproduction prohibited.

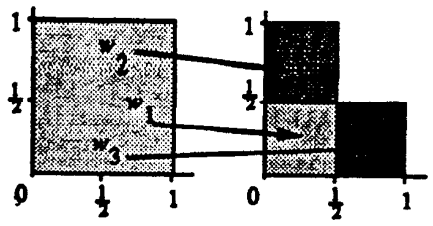

Figure 1. Three affine transformations in the plane.

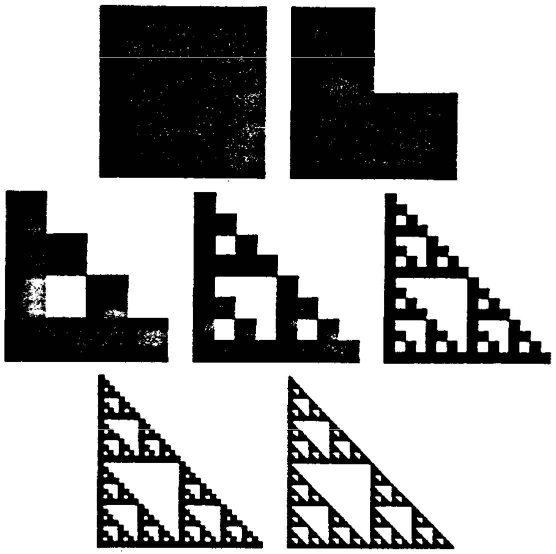

Figure 2. $A_0, A_1, A_2, A_3, A_4, A_5$, and A_∞.

A_∞ image requires 256 × 256 × 1 bit = 8192 bytes of memory. The resulting compression ratio in this example is 113.8 : 1.

It is inherently difficult to find an IFS which will encode an arbitrary set. Furthermore, in this example, the image of the Sierpinski gasket is described by a set of pixels, each being either black or white. The problem of more interest for image compression applications is the encoding of gray scale images (i.e., an image in which each pixel has many possible gray levels, not just black or white). There are two generalizations to the simple example given above which make encoding gray scale images feasible. First, instead of each w_i operating on the entire image, the w_i are restricted to operate on a section of the image. The theory of IFS's has been extended by Barnsley and Jacquin [5] to allow transforms to operate on only parts of the set rather than the entire set, in a method they call recurrent iterated function systems. The particular section, or domain, which each w_i acts on must be stored as part of the encoded image. Second, the transformations have to be generalized to three dimensions. A gray scale image can be thought of as a three-dimensional image, each pixel having an x, y coordinate, and an intensity value z. A form for the transformations which is convenient for encoding gray scale images is,

$$w_i \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & s_i \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \\ o_i \end{bmatrix}. \qquad (1)$$

Consider the sixteen transformations,

a	b	c	d	s	e	f	o
0.5	0.0	0.0	0.5	2.0	0.75	0.0	0.0
0.5	0.0	0.0	0.5	2.0	0.75	0.25	0.0
0.5	0.0	0.0	0.5	2.0	0.50	0.0	0.0
0.5	0.0	0.0	0.5	2.0	0.50	0.25	0.0
0.5	0.0	0.0	0.5	2.0	0.0	0.5	0.0
0.5	0.0	0.0	0.5	2.0	0.0	0.75	0.0
0.5	0.0	0.0	0.5	2.0	0.25	0.5	0.0
0.5	0.0	0.0	0.5	2.0	0.25	0.75	0.0
0.0	-0.5	0.5	0.0	0.25	0.25	-0.25	0.0
0.0	-0.5	0.5	0.0	0.25	0.25	0.0	0.25
0.0	-0.5	0.5	0.0	0.25	0.5	-0.25	0.25
0.0	-0.5	0.5	0.0	0.25	0.5	0.0	0.0
0.0	-0.5	0.5	0.0	0.25	0.75	0.25	0.0
0.0	-0.5	0.5	0.0	0.25	0.75	0.5	0.25
0.0	-0.5	0.5	0.0	0.25	1.0	0.25	0.25
0.0	-0.5	0.5	0.0	0.25	1.0	0.5	0.0

where the first eight transformations are restricted to act on the region $\{(x,y)|0 \le x \le 1/2, 0 \le y \le 1/2\}$, and the second eight transformations are restricted to act on the region $\{(x,y)|1/2 \le x \le 1, 0 \le y \le 1/2\}$. Similar to the example given above, the map W is defined as the union of the w_i's. Let values of $z = 0$ be represented as black, $z = 1$ as white, with intermediate values as shades of gray. The initial image A_0 is arbitrarily chosen as $z = 0.5$ for $\{(x,y)|0 \le x \le 1, 0 \le y \le 1\}$. The first six iterates, and the fixed point are shown in figure 3. In practice, the values of x, y, and z are discretised. When the image in this example is discretised as 128 × 128 pixels, and 8-bits per pixel, the encoder used in this paper [6,7] automatically encodes this image (using an equivalent set of 16 transformations) with the resulting compression equal to 356:1.

3. ENCODING AND DECODING AN IMAGE

The question that must be answered is, given an image, what is the method for finding transformations that

Reproduced with permission of copyright owner. Further reproduction prohibited.

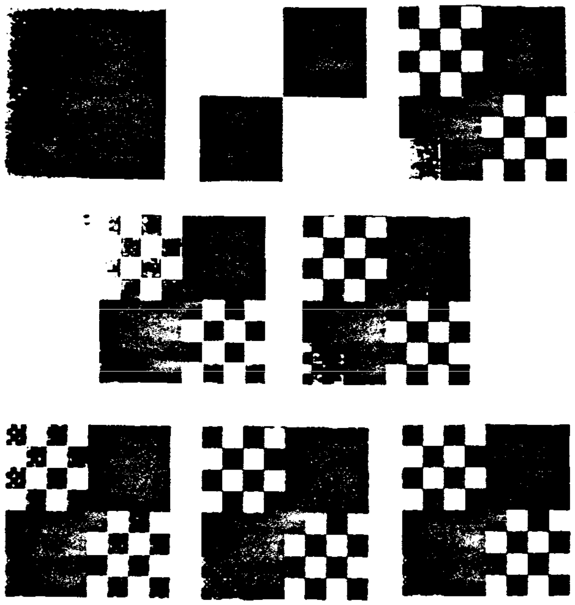

Figure 3. $A_0, A_1, A_2, A_3, A_4, A_5, A_6$, and A_∞ for the 16 transformations listed in the text.

encode it? The contractive mapping fixed point theorem suggest how to answer this question. The contractive mapping fixed point theorem guarantees that, if F is a complete metric space, and the map $W : F \rightarrow F$ is a *contractive transformation*, there exists a unique fixed point $|W| = A_\infty = \lim_{n\rightarrow\infty} W^{\circ n}(A_0)$, for any $A_0 \in F$.

Since the limit set is a fixed point,

$$|W| = W(|W|) = w_1(|W|) \cup \cdots \cup w_n(|W|). \quad (2)$$

This formula suggests how one would seek the transformations $w_1, \ldots, w_n$ which encode a given image. The goal is to have the fixed point $|W|$ approximate the desired image f. The transformations should therefore be chosen to satisfy equation 2 with $|W|$ replaced by f, i.e., the transformations, when applied to f, should result in f. The $w_1(f), \ldots, w_n(f)$ are said to *cover* the image f. Referring back to the two examples in the previous section, it is seen that, given the Sierpenski triangle, or the fractal square pattern, by satisfying equation 2 the transformations encoding these images could be found.

In the two examples, the covering $W(f)$ is exact. Given an arbitrary set f, it is not possible in general to exactly cover f with a finite number of transformations of itself. The obvious question is then: what happens if the covering $W(f)$ is approximate? A corollary of the contractive mapping fixed point theorem, which Barnsley calls the Collage Theorem, puts a bound on the error between $|W|$ and f when $W(f)$ does not exactly equal f. The theorem says that the closer the covering $W(f)$ is to the original set f, the closer the fixed point $|W|$ will be to f, and that this is especially true if the transformations composing W are very contractive.

In figure 4 part of the encoding process for this image is illustrated. The figure demonstrates how one section of the image, called a range (R_i), is covered as closely as possible by applying a transformation w_i to a domain (D_i). To complete the encoding process, a w_i and D_i must be found to cover each R_i, and the R_i's must completely tile the image. To facilitate compact specification of the transformations, the sets from which D's and R's are chosen are restricted to be geometrically simple, and limited in number. The w_i's must be chosen such that upon iteration, a fixed point is reached. In light of the collage theorem, it is surprising that when the map W is constructed, it is not necessary to impose any contractivity conditions on the individual transforms. The necessary contractivity requirement is that W be *eventually contractive* [8]. A map $W : F \rightarrow F$ is eventually contractive if there exists a positive integer m such that the m^{th} iterate of W is contractive. Note that in the gray scale example of section 2, half of the transformations are not contractive in the z direction.

As shown for the simple examples in the previous section (figures 2 and 3), decoding an image is performed by starting with an arbitrary initial image, and iterating the transformations until the fixed point is reached. This process is shown for an encoding of "tank farm" in figure 5. The compression of this image was 8.66:1, and the PSNR $= -20\log\left(\frac{\text{rms error}}{255}\right) = 33.6$ dB.

4. APPLICATION TO DTED

Application of image compression to geographic map data is of particular interest to the Navy. Geographic map data comes in a variety of formats, and there has been extensive work done in compressing map databases for various applications. In this section, the problem of compres-

Reproduced with permission of copyright owner. Further reproduction prohibited.

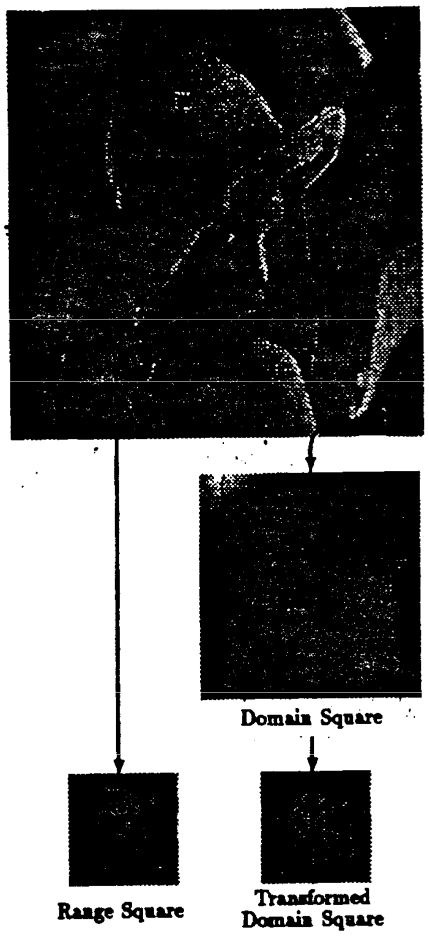

Figure 4. Part of the encoding process.

sion of the Digital Terrain Elevation Database (DTED) is addressed. The database consists of elevation data for a grid of longitude–latitude coordinates where the grid points are roughly 100 meters apart. A complete description of DTED can be obtained from the Defence Mapping Agency. Alward and Nicholls [9] examined hierarchical data structures as applied to DTED. The data structures result in some data compression, although compression was not the primary goal of that investigation. In this section, the problem of interest is evaluating the performance of the iterated transform (IT) method on DTED. This would be useful for applications where data compression is the primary concern, and issues such as hierarchical structure, access time, and decoding time are of secondary importance. As a means of evaluation, the DTED images

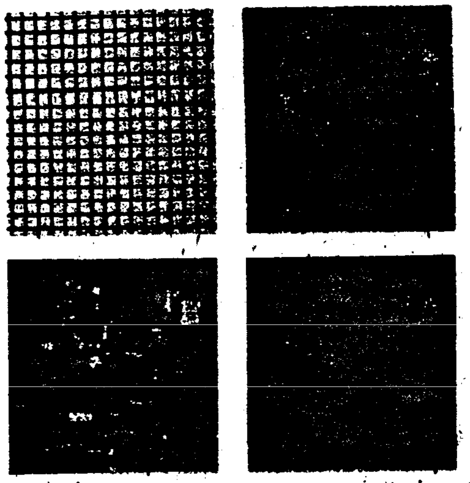

Figure 5. Initial image, first iterate, second iterate, and tenth iterate for an encoding of the "tank farm" image.

were also compressed with an ADCT and mean residual vector quantization algorithm (MRVQ). DTED data can be thought of in terms of a gray scale image where the longitude and latitude identify the pixel, and the elevation is the pixel value. For the purpose of possible Navy application, the data were transformed from their original linear scale between 0 and 10000 meters, to a logarithmic scale between 0 and 255. The quantization results in a compression from 14 bits per datapoint to 8 bits per datapoint. This logarithmic scale, shown in figure 6, represents lower elevations more accurately than higher elevations, the rationale for this being that lower elevations areas are more likely to be important for Navy applications, and that nearly all of the earth's surface (particularly near coastlines) is at relatively low elevation.

The iterated transform algorithm was identical to that used in reference 6 except for one significant modification. The algorithm was modified to encode sections of the image on coastlines with increased accuracy. For image sections that contained coastline, the error criteria was tightened. This resulted in more segmentation, and therefore higher fidelity in these areas. The ADCT algorithm used was similar to that described by Chen and Pratt [10], except for a modification similar to that described above for the iterated transform algorithm. To encode coastlines more accurately, the decision level quantiser table was compressed for sections of the image on coastlines. Improving the fidelity at the coastlines resulted in a

Reproduced with permission of copyright owner. Further reproduction prohibited.

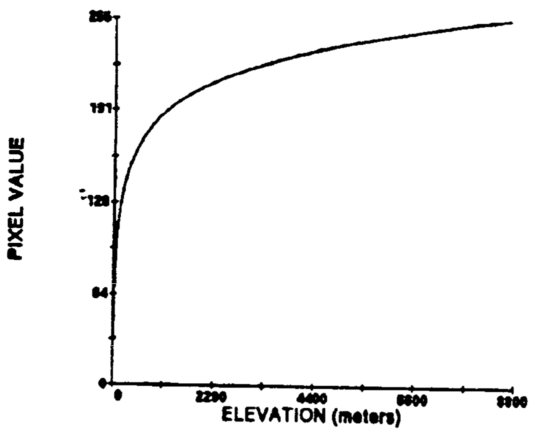

Figure 6 . The scale transformation used for the DTED data.

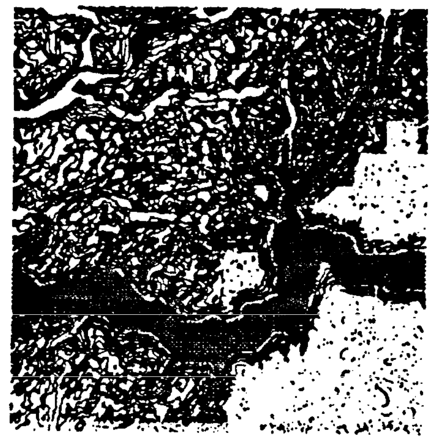

Figure 7 . Pixels of value 100 ±2 meters.

Table 1 . Results for encodings of figure 8.

Method	Compression	PSNR(dB)	Figure
IT	44.86:1	32.98	
IT	21.49:1	35.08	9
ADCT	47.33:1	30.51	
ADCT	21.08:1	34.92	10
MRVQ	32.00:1	31.36	11

Table 2 . Results for encodings of figure 12.

Method	Compression	PSNR(dB)	Figure
IT	104.11:1	34.09	
IT	42.08:1	39.03	13
ADCT	75.53:1	33.86	
ADCT	43.76:1	38.14	14
MRVQ	32.00:1	36.93	15

decrease in the overall compression-fidelity performance. The MRVQ algorithm used was based on the method described by Linde et al. [11]. Codebooks were generated from 2 sections of DTED similar to (but not including) the section tested.

The compression methods being considered are properly applied to images with a dynamic range appropriate to the number of bits used to store the image, and a relatively smooth distribution of gray level intensity values over this range. In figure 7, the 512 × 512 section of DTED, which is 1° east of that section shown in figure 8 is displayed where sea level is displayed in gray, all pixels with an elevation that is a multiple of 100 (± 2) are displayed in black, and all other elevations being displayed in white. It appears that the majority of the data contained in this section of DTED were created from 100-meter contour maps, the result being that a disproportionate number of datapoints are at multiples of 100 meters. In the southeast corner of the map, it can be seen that the number of points at 100 meters are far more dispersed. In this section of the map, roughly the number of datapoints that would be expected based on random elevations are present. The biased quantization illustrated by figure 7 is evident in other sections of DTED.

Because of this biased quantization, the reconstructed image resulting from compressed encodings will have a smoother distribution of pixel values than the original image. This will lead to an artificially poorer measured fidelity of the encoded images. In areas of DTED where the data are "properly" digitized, this situation will not occur. Because not all the data are quantized at a course resolution, but only most of it, taking advantage of this quantization is not simple. Although it has not been done here, before applying lossy compression techniques such as iterated transforms, ADCT, or VQ, requantization of the data in such a way that the majority of the datapoints retain their correct values could result in overall improved performance.

Tests in this section were performed on two 512 × 512

Reproduced with permission of copyright owner. Further reproduction prohibited.

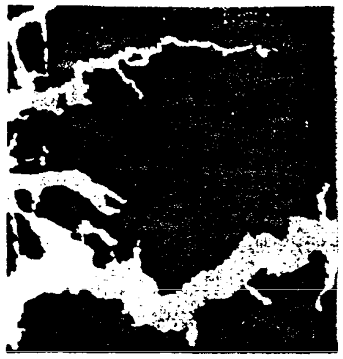
Figure 8 . Original 5° to 5° 25.6' E, 61° to 61° 12.8' N.

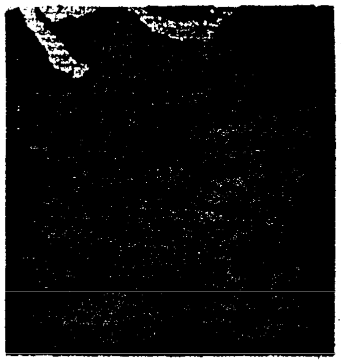
Figure 12 . Original 6° to 6° 25.6' E, 61° 12.8' to 61° 25.6' N.

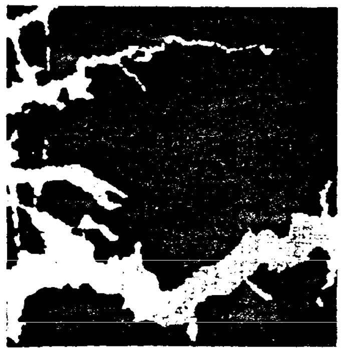
Figure 9 . The decoded IT image of figure 8.

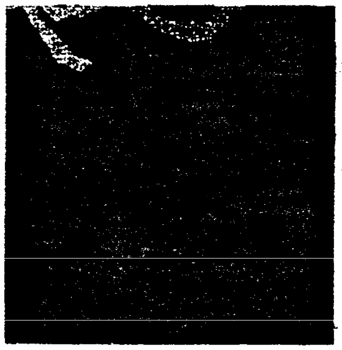
Figure 13 . The decoded IT image of figure 12.

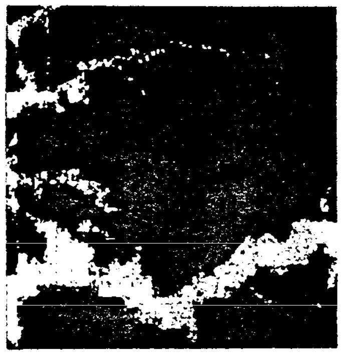
Figure 10 . The decoded ADCT image of figure 8.

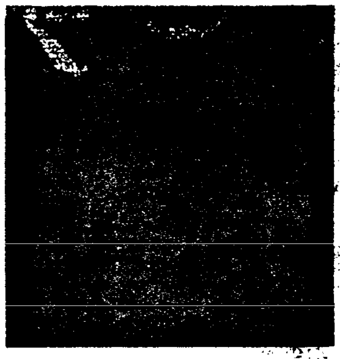
Figure 14 . The decoded ADCT image of figure 12.

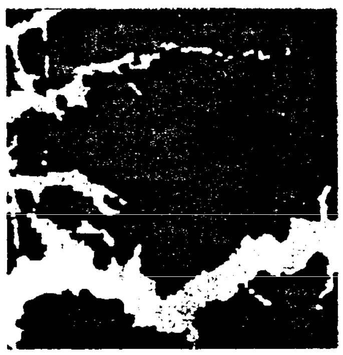
Figure 11 . The decoded MRVQ image of figure 8.

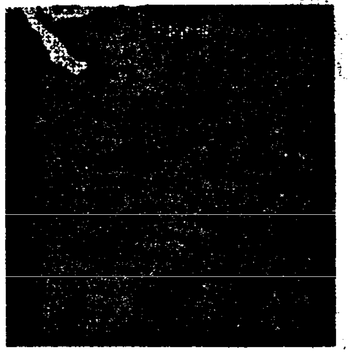
Figure 15 . The decoded MRVQ image of figure 12.

Reproduced with permission of copyright owner. Further reproduction prohibited.

sections located in the fiords of Norway. These test images are shown in figures 8 and 12. Figure 8 covers the section of the earth from 5° to 5° 25.6' east and 61° to 61° 12.8' north and figure 12 the section from 6° to 6° 25.6' east and 61° 12.8' to 61° 25.6' north. These sections of DTED were chosen because the topology of the fiords served as a severe test of the fidelity of the compression methods. A more thorough study where a broad area of the database containing a representative amount of flat and mountainous regions would be necessary in determining the compression and fidelity possible for the complete database.

Figures 9, 10, and 11 show gray scale reconstructed images from typical encodings of the image in figure 8 using iterated transforms, ADCT, and MRVQ respectively. Similarly, figures 13, 14, and 15 show reconstructed images from encodings of the image in figure 12. The fidelity and compression of these (and other) encodings are summarized in tables 1 and 2.

5. CONCLUSIONS

The compression versus fidelity results indicate that the iterated transform algorithm performed well when compared with the ADCT and MRVQ methods. Other important factors to consider when comparing these compression algorithms are access time, decoding time, and encoding time. Iterated transforms, along with ADCT are variable bit-rate methods, which would result in slower access times than the fixed bit rate MRVQ algorithm. Iterated transform encoding requires an extensive search procedure, making it slower than MRVQ, which also requires a search, albeit a shorter one. Iterated transform encoding is also slower than ADCT, which requires only a transformation and quantization. For most applications, encoding would be a one-time procedure; therefore, encoding time would not necessarily be an important concern. If an application required all or a large fraction of DTED be encoded, then the computer costs for encoding become significant. For many applications, the speed of decoding an image might be a critical requirement. The decoding for iterated transforms is a simple iteration, making it faster than ADCT (where decoding takes as long as encoding), and slower than MRVQ, which is essentially a table lookup.

6. REFERENCES

[1] A.E. Jacquin, "*A Fractal Theory of Iterated Markov Operators, with Applications to Digitial Image Coding,*" Ph.D. Thesis, Department of Mathematics, Georgia Institute of Technology, (1989).

[2] A.E. Jacquin, "Fractal Image Coding Based on a Theory of Iterated Contractive Image Transformations," *SPIE Visual Comm. and Image Processing '90*, Vol. 1360, pp. 227-239, (1990).

[3] J.E. Hutchinson, "Fractals and Self-Similarity," *Indiana University Mathematics Journal*, vol. 35, p. 5, (1981).

[4] M.F. Barnsley, *Fractals Everywhere*, Academic Press, Inc., San Diego, CA, (1988).

[5] M.F. Barnsley and A.E. Jacquin, "Application of Recurrent Iterated Function Systems to Images," *SPIE vol. 1001, Visual Comm. and Image Processing*, p. 122, (1988).

[6] E.W. Jacobs, Y.Fisher, and R.D. Boss, "Image Compression: A Study of the Iterated Transform Method" *to be published in Signal Processing*, vol. 29 no. 3.

[7] R.D. Boss and E.W. Jacobs, "Studies of Iterated Transform Image Compression, and its Application to Color and DTED," NOSC TR 1468, December 1991.

[8] Y. Fisher, E.W. Jacobs, and R.D. Boss, "Fractal Image Compression Using Iterated Transforms," submitted for publication in *Data Compression*, J.A. Storer (ed.), Kluwer Academic Publishers, Norwell, MA, (1991).

[9] Alward, H.L. and D.A. Nicholls, "Hierarchical Data Structures for a Digital Terrain Map System," AFWAL-TR-86-1177, (1986), available from DTIC.

[10] W. Chen and W.K. Pratt, "Scene Adaptive Coder," *IEEE Transactions on Communications*, vol. 32, pp. 225-232, (1984).

[11] Y. Linde, A. Buzo, and R.M. Gray, "An Algorithm for Vector Quantizer Design," *IEEE Trans. on Comm.*, vol. COM-28, no. 1, pp. 84-95, (1980).

Reproduced with permission of copyright owner. Further reproduction prohibited.