(1)

Fractal Image Encoding: SBIR Phase I Final Report NETROLOGIC, Inc.

November 17, 1990

Submitted to: Dr. Michael Schlesinger
Office of Naval Research

Yuval Fisher
UCSD Supercomputing Facility
University of California, San Diego
La Jolla, CA 92093
and
NETROLOGIC, Inc.
5080 Shoreham Place, Suite 201
San Diego, CA 92122

Albert Lawrence, Consultant
NETROLOGIC, Inc.
5080 Shoreham Place, Suite 201
San Diego, CA 92122

Table of Contents

Executive Summary

The coding, storage, and reconstruction of images is a major concern in the application of computer technology to technical and scientific problems. One example is the flood of geophysical and intelligence data originating from satellite platforms. In such applications it is highly desirable to reduce the storage and transmission requirements for image data. An image can be coded compactly when it is possible to exploit self similar redundancy in the image. The development of such a so-called "fractal" method for compressing image data has been the focus of our research project.

Our approach to image compression has been to tessellate the image with a tiling which varies with the local image complexity, and to check for self similarity amongst the tiles. Self similarities are coded as systems of affine transformations which can be stored far more compactly than the original image. This method is inherently lossy, since the self similarities are never exact. Although the tiling technique yields good results in many cases, we have also begun to investigate contour schemes which may lead to irregular tilings with even better compression ratios, computation time and signal-to-noise ratios.

We have tested our encoding scheme on a variety of test images, gaining compression ratios greater than 40:1. At high compression ratios, the scheme yields better signal to noise ratios than are reported for other techniques. Our scheme is versatile in that it allows a trade off between compression, reconstructed image fidelity and encoding time. Our methods are computationally intensive but are feasible for non-real time applications on workstations or main frame computers. The algorithms can be accelerated considerably by dedicated hardware for real time requirements.

Fractal compression is a promising approach to image compression. Within a very short development time, fractal techniques have yielded results which rival the best examples of data compression afforded by other methods. Although fractal encoding of images is complex and may require specialized hardware for real time applications, the decoding process can be widely utilized because it is simple, fast, and suitable for software implementation. Thus, it can be run on workstations or personal computers without special requirements.

We recommend further research and development along the following lines. First, the tiling scheme is sufficiently mature to consider hardware implementation for possible real time applications. Second, we expect that the application of fractal techniques with other image processing methods, such as contour detection, will lead to even better results. Finally, research into the mathematical foundations of the subject is warranted. We believe that a program which integrates hardware engineering, software development and further mathematical research will yield the best results in the long run.

Section 1.0. Introduction.

The objective of the Phase I research was "to improve [lossy] image compression techniques which are based on fractal constructions." This was an ambitious undertaking, since at the time of the proposal no such techniques were widely known. In fact, although claims about "fractal image compression" existed, there were no bench mark results and no hard data on such schemes.

Thus, the Phase I research set out to prove the feasibility of such a scheme, and the conclusion of this research effort is: such a scheme is feasible. We have implemented a working, completely automatic image compression algorithm. The algorithm yields compression results that compare favorably to other state of the art techniques.

The work of the last six months at NETROLOGIC can be broken down into two parts. The first part demonstrates automatic encoding of digital image data by storing an approximation of the image as a collection of affine transformations of the plane. The amount of memory required to store the transformations is considerably smaller than the amount of memory required to store the original image, and the transformations can be converted back to an image by a simple and fast procedure.

The second part of the work deals with alternative schemes to compress images, some transform based and some not. Since the first part already demonstrated our ability to encode images as transformations, the results from this work are beyond the scope of the original Phase I proposal. Since these schemes are preliminary, they should be considered proprietary. Like the technique in the first part, these techniques are based on completely new algorithms developed at NETROLOGIC and the University of California, San Diego.

One important note remains to be made: Although the work demonstrates the feasibility of encoding images as transformations, the theoretical foundation for the subject is requires more development. The major theorem in the field is the contractive mapping fixed point theorem, which is unsatisfactory because the bounds it gives on certain rates of convergence do not come close to the actual convergence rates. This suggests that we do not understand the mechanism that controls the degree to which an image can be encoded. Until the subject has a well formulated and successful theoretical foundation, the results will continue to be largely empirical and less than satisfactory.

Section 2.0. Background.

The following is a brief history of the development of the subject. Hutchinson [5] introduced the theory of iterated functions systems (a term coined by Barnsley) to model self similar sets (such as in figure 3). Demko, Hodges, and Naylor [6] first suggested using iterated function systems to model complex objects in computer graphics. Barnsley, Demko, Elton, Sloan and others generalized the concepts and suggested the use of fractals to model "natural scenes". In his thesis [7], A. Jacquin developed an image encoding scheme based on iterated Markov operators

on measure spaces and used it to encode 6 bit/pixel monochrome images.

M. Barnsley (see [2] or [3]) is credited with stimulating interest in image compression based on "fractal" techniques. While much effort has been spent on the use of fractals to generate images, Barnsley popularized the converse notion of encoding the content of images using fractals. Although many claims were made in this field recently, results based on rigorous test procedures are almost nonexistent. In an effort to bring results and details to scientific scrutiny, work was begun at the University of California, San Diego (UCSD), the Naval Ocean Systems Center (NOSC), in addition to NETROLOGIC. The result of the work at NETROLOGIC and the current research is presented in section 4.0.

The subject is still in its infancy - the most important observation is that the theoretical foundation is still very weak and that the underlying mechanism driving the encoding process is not well understood. The existing theoretical foundations can be found in the next section.

Section 3.0. Theory.

In this section we motivate the use of affine maps to encode images. The main tool of the subject is an old but powerful theorem, the contractive mapping fixed point theorem. First we need a definition.

Definition. *Let δ be a metric on a space F. A map $W : F \to F$ is said to be a contraction if there exists a positive real number $s \leq 1$ such that*

$$\delta(W(x), W(y)) < s\delta(x, y)$$

for any two points $x, y \in F$. If $s < 1$ then W is said to be a strict contraction.

Theorem (Contractive Mapping Fixed Point). *Let F be a complete metric space with metric δ. If $W : F \to F$ is a contraction, then there exists a unique point $g \in F$ such that $g = W(g)$. Moreover, for any $f \in F$, the fixed point is the limit $g = \lim_{n\to\infty} W^{\circ n}(f)$.*

In the next section we apply this theorem to produce a simple example of image compression.

Section 3.1. A Simple Example.

The following example shows a simple application of the contractive mapping fixed point theorem. In this example F is the space of compact subsets of R^2 and δ is the Hausdorff metric, then (F, δ) is a complete metric space. (The exact definition of the Hausdorff metric is not important, it is sufficient to think of it as measuring the extent to which two sets in the plane overlap). We can then define the three transformations shown in figure 1:

$$w_1 \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$
$$w_2 \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{2} \end{bmatrix}$$
$$w_3 \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \frac{1}{2} \\ 0 \end{bmatrix}.$$

For any set S, let

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

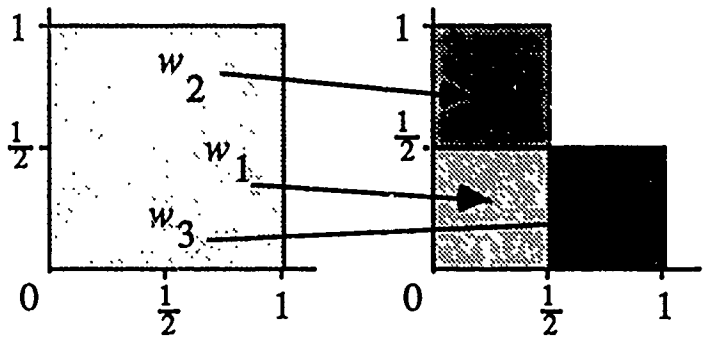

Figure 1. Three affine transformations in the plane.

Consider the following process: begin with the square $A_0 = \{(x,y) : 0 \leq x \leq 1, 0 \leq y \leq 1\}$, as shown on the left of figure 1. Let $A_1 = W(A_0)$ and in general $A_n = W(A_{n-1})$. Figure 2 shows A_1, A_2, A_3 and A_4. The sets appear to be converging to a limit set shown in figure 3. In fact, the maps w_i are strictly contractive in the euclidean metric, and it is not hard to show that this implies that the map W is contractive in the Hausdorff metric. As $n \to \infty$, the sets A_n converges (in the Hausdorff metric) to a limit set A_∞, shown in figure 3. Moreover, for any compact set $S \subset R^2$, $W^{\circ n}(S) \to A_\infty$ as $n \to \infty$.

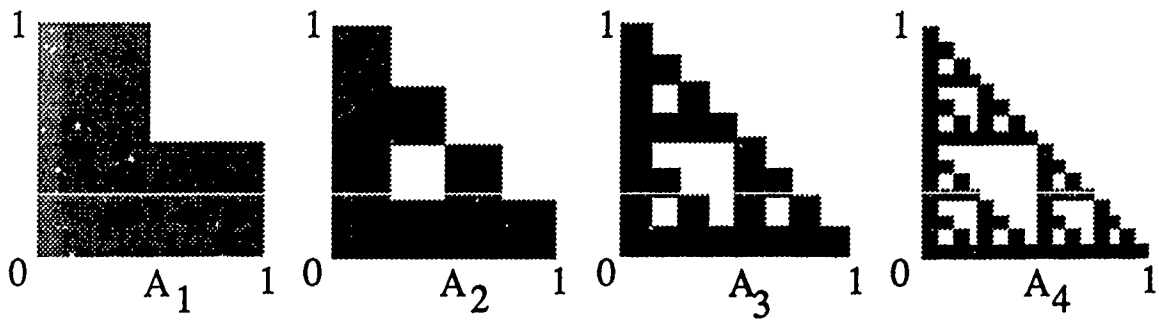

Figure 2. $A_1 = W(A_0)$ and its images A_2, A_3, and A_4.

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. Also, it is not difficult to see why such a thing is true. The w_i are contractive; they halve the diameter of any set to which they are applied. Thus, the images of any initial A_0 will shrink to points in the limit as the w_i are repeatedly applied.

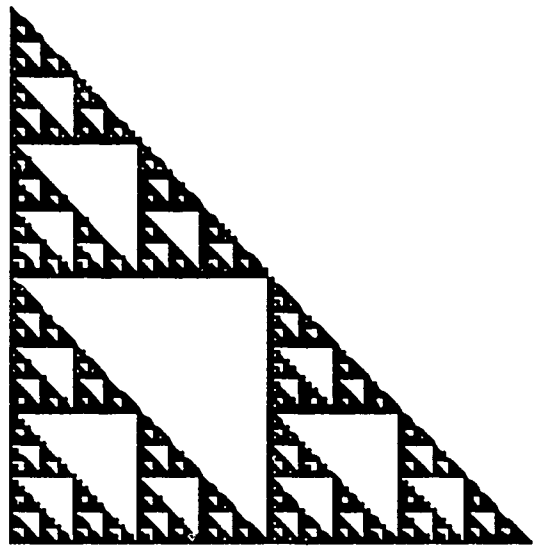

Figure 3. The limit set $A_\infty = \lim_{n\to\infty} W^{\circ n}(A_0)$.

Each w_i is determined by 6 real values, so that for this example we require 18 floating point numbers to define the image. In single precision, this requires 72 bytes. The memory required to store an image of the set depends on the resolution; figure 3 requires $256 \times 256 \times 1bit = 8192$ bytes of memory. The resulting compression ratio in our particular example is thus 114 : 1. However, in this particular example, the limit set is a fractal and can thus be decoded to any resolution. The resulting compression ratio can be honestly said to be infinite. In applications, however, it is less than honest to decode an image at a different size than it was encoded at and to then claim very high compression rates. Just as with the fractal in this example, the fractal compression scheme we describe later will generate detail at all scale levels, even though such detail is not present in the original image. *It is very important to include both the original image size and the signal to noise ratio (or some other measure of error) when giving results for any image encoding method and "fractal" image encoding methods in particular.*

Section 3.2. The Space of Images.

The space in which we work when compressing images can be defined in the following way. We denote the closed interval $[0, 1]$ by I, and the n-fold Cartesian product of I with itself by I^n. Let F be the space consisting of all graphs of real Lebesgue measurable functions $z = f(x, y)$ with $(x, y, f(x, y)) \in I^3$. Thus f is bounded. We think of a point in F as an abstract image of infinite resolution, with $f(x, y)$ representing the grey level (with 0 being black and 1 being white)

at the point (x, y) in the image. We can model images with finite resolutions by partitioning I^2 with a rectilinear grid and either insisting that f be constant on the boxes of the grid, or by averaging f over each box. The model with infinite resolution allows us to handle the theory somewhat more clearly. Color images can be encoded as graphs of functions $f : I^2 \to I^3$ with range points representing the color model of choice, for example RGB values.

We metrize F using the metric induced by the essential supremum:

$$||f||_\infty = \inf\{\alpha : \mu(f^{-1}((\alpha, 1])) = 0\}.$$

The metric we wish to use is

$$\delta(f, g) = || \, |f - g| \, ||_\infty.$$

The space F with the metric δ is complete, allowing us to use the contractive mapping fixed point theorem, when we identify images which have distance 0. Other spaces of images are possible, for example the space of positive Borel measures supported on I^2, but this space is difficult to metrize, and the particular space and metrization is of little practical consequence ultimately. This is especially true in light of the inability of the theory to bound the converges rates even weakly, as we show later.

We now return to the application of the contractive mapping fixed point theorem to the problem at hand. Let $W = \cup_{i=1}^n w_i : F \to F$ denote some contractive map, which we assume is built up of a union of local maps $w_i : F \to F$ (as in equation 1, for example).

Following Hutchinson's notation [5], we denote the fixed point $g = |W| = \lim_{n \to \infty} W^{\circ n}(f)$. Then

$$|W| = W(|W|) = \bigcup_{i=1}^{n} w_i(|W|). \tag{2}$$

We say that W encodes an image $f \in F$ if $f = |W|$. Given W, it is easy to find the image that it encodes – begin with any image f_0 and successively compute $W(f_0), W(W(f_0)), \ldots$ until the images converge to $|W|$ (just as in the example in section 3.1). The converse is considerably more difficult: given an image f, how do we find a mapping W such that $|W| = f$? We know of no general, non-trivial solution to this problem, nor do we expect that one exists. We attempt instead to find an image $f' \in F$ such that $\delta(f, f')$ is minimal with $f' = |W|$. Equation 2 suggests how this might be possible. We seek domains $D_1, \ldots, D_n \subset F$ and corresponding transformations $w_1, \ldots, w_n : F \to F$ such that

$$f \approx W(f) = \bigcup_{i=1}^{n} w_i|_{D_i}(f). \tag{3}$$

This equation says: cover f with parts of itself; the parts are defined by the D_i and the way those parts cover f is determined by the w_i. Equality in equation (3) would imply that $f = |W|$. Since we cannot cover f exactly with parts of itself, we try to do the best we can and hope that $|W|$ and f will not look too different, i.e. that $\delta(|W|, f)$ is small. The following observation, which is due to Barnsley [1] and which he calls the Collage Theorem, gives us hope that this can be done. It is a corollary of the contractive mapping fixed point theorem.

Corollary. *Let $W : F \to F$ be a contraction with contractivity s and let $f \in F$ be an image. Then $\delta(|W|, f) \leq \frac{1}{1-s}\delta(W(f), f)$.*

Our problem is to find a W such that $\delta(W(f), f)$ is minimized and such that s is small. We will then know that $|W|$ is close (in δ) to f. However, the bound in the corollary is not very good; it provides motivation only and not a useful bound in practice. In fact, it is possible to generate examples in which the bound in the corollary is arbitrarily large while $\delta(|W|, f)$ is bounded. We have found empirically that restricting s to be small results in poorer encodings, because while the $\frac{1}{1-s}$ term decreases, $\delta(W(f), f)$ increases. Table 1 demonstrates this phenomenon.

Remark: We can *always* approximate any given image f to within any $\epsilon > 0$. Since the simple functions (functions whose range consists of a finite number of points) are dense in F, we can find simple functions $g_1, \ldots, g_n$ such that $\delta_p(\sum g_i, f) < \epsilon$. For details see [8]. We can then construct maps $w_1, \ldots, w_n$ and a map W using the simple functions g_i. This is not a deep point, because we sacrifice compression in order to achieve accuracy – that is, we require a large number of maps $w_1, \ldots, w_n$ in order to have $\delta(f, W(f))$ small, and n grows rapidly as $\epsilon \to 0$.

We found, while proving the convergence of W, that contractiveness was not essential. It is sufficient for W to be eventually contractive, meaning that there exists some positive integer m such that $W^{\circ m}$ is contractive. We generalized the corollary to give a new bound on the efficacy $\delta(|W|, f)$ of the encoding W.

Generalized Corollary. *For $f \in F$, $W^{\circ m} : F \to F$ contractive with contractivity $\sigma < 1$, and s_{max} the expansiveness of W,*

$$\delta(|W|, f) \leq \frac{1}{1-\sigma}\frac{1-s_{max}^m}{1-s_{max}}\delta(W(f), f).$$

In encoding images, we restricted ourselves to eventually contractive maps, since they yield better results. Nevertheless, the theoretical bound on the efficacy of the encoding $\delta(|W|, f)$ in both the eventually contractive and contractive cases is poor, as demonstrated in table 1. The metric used in this table is the rms metric.

Table 1 gives results for a typical portrait image. The first two entries in the table are encodings resulting from contractive maps. The third entry is an encoding with an eventually contractive map. The table demonstrates two points. First, the eventually contractive map gives the best encoding. Second, even though restricting the contractivity of W improves the bounds the contractive mapping

theorem corollaries, it worsens the encoding. This demonstrates that the bounds in the corollaries are poor; hence they provide motivation rather than actual bounds.

$\delta(W(f),f)$	$\delta(\|W\|,f)$	s_{max}	σ	m	Col2-Col1	Theorem
21.965	23.487	0.8	0.8	1	1.513	109.825
20.306	20.976	0.9	0.9	1	0.67	203.06
18.937	19.621	4.05	0.85	5	0.684	720874.8

Table 1

In the implementation of the algorithm, we restrict ourselves to affine maps of the form

$$w_i \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & s_i \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \\ o_i \end{bmatrix} \tag{4}$$

where the $a_i, b_i, c_i, d_i, e_i, f_i, s_i$, and o_i are determined by minimizing $\delta(W(f), f)$. Finding good values for these parameters is the crux of the problem, and we describe a method to find such values in the next section.

Section 4.0. Results.

This section contains the main Phase I research results. It is organized into two parts; the first demonstrates the feasibility of encoding digital images as a collection of "tile" transformations. This scheme is called *tiled transform image encoding*. The transformations can be constructed into an image which is an approximation of the original image (the compression scheme is lossy), and there is a trade off between compression fidelity, the extent to which the reconstructed image resembles the original, and compression ratio, the ratio of memory required to store the original image to the memory required to store the compressed image. Reference [1] gives further details.

The second part of this section exhibits results obtained from alternative image compression schemes, also developed during the Phase I research. These schemes are not as mature as the tiled transform encoding scheme, but they do display several attractive features. First, they provide a foundation for an alternative method of encoding images as transformations (this alternative has not yet been studied). Second, the schemes are simple and fast to implement and execute. And finally, the schemes can be implemented in a variety of ways, utilizing standard compression techniques or new "fractal" based techniques. These algorithms can be loosely described as contour based, since they depend on extracting contours (such as level curves or edges) from an image.

Section 4.1. Results from Phase I.

A detailed explanation of the results and the image compression scheme may be found in [1]. This text informally explains the general idea of the scheme. The

compression scheme depends on the following premise: Given an image, for example a face, it is reasonable to expect that small portions of the image, for example the tip of the nose, resemble larger scale features, such as a properly scaled and rotated chin. When small scale features can be represented as transformations of larger scale features, an image can be stored as the large scale features plus the set of transformations needed to define the small scale features. Since it essentially never happens that a tip of a nose looks exactly like a skewed chin, the resulting image is never identical to the original. On the other hand, the amount of memory required to store an image in this different way is often much less than the memory required to store the original.

An image is digitally stored as a collection of values. Each value represents a grey level, for example 0 may be black and 255 may be white, with the values interpolating the grey level between these extremes. Successive values represent dots or pixels, forming a matrix which can be viewed on a monitor and which looks like an image, since the human visual system tends to ignore the pixelization.

The main difficulty of the scheme described above is finding the transformations. Our tiling transformation image encoding scheme (TTIE) finds the transformations in the following way. The images is broken up into tiles; for example, a 256×256 pixel image is partitioned into contiguous, non-overlapping 8×8 tiles, called range tiles. The image is also partitioned into 16×16 pixel tiles called domain tiles. For each range tile, we seek a domain tile which looks most like it. The computer determines how much two tiles "look alike" by an affine least squares fit of the pixel values, using 1 of every 4 pixels to accommodate the difference in domain to range tile size. Domain tiles are checked in 8 possible orientations, corresponding to the symmetries of the square. The transformations stored are then determined by which domain mapped to which range, the orientation of the domain, and the affine transformation (scaling and offset) on the grey levels of the domain.

Once all range tiles have been covered the image can be reconstructed in the following way. An arbitrary initial image is chosen, for example an image consisting entirely of zeros. The part of the initial image which corresponds to the domain tile is copied to a separate storage location. This subimage is oriented properly, as determined by the transformation. Its pixel levels are scaled and offset. It is shrunk by taking only one of every 4 pixels, so that it shrinks to size 8×8. And it is then put in the position of the range tile determined by the current transformation. The domain tile is unaffected. All of the transformations are applied. This whole cycle is repeated several times until the resulting image remains stable. Convergence is a consequence of the contractive mapping fixed point theorem, as long as the affine transformations are taken to be contractive (or eventually contractive).

The transformations completely determine the final image, independently of the initial image chose. There is no need to store the large scale detail separately; it is created out of the small scale detail stored in the 8×8 tiles. The small scale detail is then created in turn by the mapping of the large scale detail into the 8×8 tiles.

The algorithm is actually considerably more sophisticated than the one outlined

above. The size of the domain and range tiles is not fixed. Rather, it varies with the local image complexity in order to allow fewer transformations to be used in relatively flat portions of the image and more transformations to be used in complex regions of the image. Also, the search through the domain tiles is not exhaustive, since this is computationally intensive. Rather, the domains and ranges are classified and only the domains of the same class type as a given range are searched for a good least squares fit.

Figure B.1 shows the decoding process. The initial image is, in this example, a pattern of vertical lines, chosen to show the structure of the transformations after one application of all of the maps. The figure contains, first from left to right, top then bottom: The initial image, one application of all of the transformations, two applications of the transformations, and ten applications of the transformations by which we have converged. The initial image is shown in figure B.2. The last image in figure B.1 can be encoded in 1/10th the memory required to store the image in figure B.2, with an rms error of 8.59 (a signal to noise ratio of 29.5db).

Table 2 below gives several typical results for several images. The table specifies the image size in pixels; the compression in memory required to store the original image vs. the memory required to store the compressed image; the signal to noise ratio; the cpu seconds of computation time required to encode the image on a Convex C210; and the number of transformations in the encoding.

In general, larger images yield better compression ratios, since there is more interpixel correlation. The compression scales roughly with the image size, however, so that a 512×512 image, which contains four times as much information as a 256×256 image, will have roughly four times the compression ratio.

Image	Size	Comp	SNR (db)	Time (sec)
Lena	512×512	15.6	32.1	899.9
Lena	512×512	38.7	29.2	-
Lena	512×512	23.5	30.0	2582.0
Lena	256×256	11.3	28.8	234.6
City	256×256	7.1	26.6	530.3
Collie	256×256	19.5	30.2	1898.9

Table 2

Figure B.3 shows the original 256×256 pixel collie image and three compressed versions (left to right, top to bottom) at respective compressions of 63.0:1 with a signal to noise ratio of 25.2db, 28.2:1 with a signal to noise ratio of 29.3db, and 16.6:1 with a signal to noise ratio of 30.4db. Figure B.4 shows an encoding of a 512×512 original Lena at compression 38.7 with a signal to noise ratio of 29.2db.

The images shown in appendix B have not been postprocessed. With postprocessing, most of the boxy artifacts introduced by the compression scheme can be eliminated. Some more details can be found in appendix A.

It is important to stress that the software yielding these results has not been optimized. It is of a highly developmental nature, resulting in somewhat less than

optimal results. Never the less, the results are comparable to other image compression techniques (See section 5.0).

Section 4.2. Extra Results: Most Promising Lines for Future Research.

This section describes results which are not directly related to the Phase I proposal, in the sense that they do not demonstrate the feasibility of "fractal" image compression. Because of the preliminary nature of the research, this section should be considered proprietary.

The results are all based on various manipulations of contours. The schemes are:

- Transform encoding of contours (TEC);
- Image encoding through level curves;
- Image encoding through leveling of edges.

Section 4.2.1. Transform Encoding of Contours.

We briefly discuss the one dimensional analogue of the two dimensional tiled transform image encoding scheme. The image, in this case, becomes a contour, and the tiles become portions of the contour called chains. The algorithm is still the same, however: The contour is partitioned into long domain chains and short range chains (the particular lengths are not important and can be varied to alter the final compression and the fidelity of the reconstructed contour). For each range chain, a domain chain is found which minimized the root mean square distance between the range chain and an affine transformation of the domain chain. These transformations are recorded, along with the endpoints of the range chains. An approximation of the original contour can then be reconstructed from these transformations by iteratively applying all of the transformations. For more details, see [4].

Section 4.2.2. Image Encoding Through Level Curves.

The idea of using level curves to encode an image arose in conversation with Josh Deutsch, of the physics department at the University of California, Santa Cruz. Spencer Menlove, of NETROLOGIC, also contributed to both the implementation and design of these algorithms.

A level curve is a contour in the image which has a constant grey level. At sharp contrast points of the image, the contour is forced to pass through pixels which may not be of constant grey level but which approximate the position of the contour had the image been of infinite resolution and the grey levels been continuous. Level curves are extracted at several different grey values. These level curves can be stored compactly, using, for example, the transform encoding of contours method described in the previous section. To reconstruct the image, the grey levels of pixels between contours are interpolated.

This scheme is conceptually simple, simple to implement, not computationally intensive, and moderately successful.

Due to time constraints, this scheme and the TEC scheme were not united, so the results reflect compression of the date using a lossless standard technique. From the experimentation with the TEC scheme, we hope to achieve an improvement in compression by roughly a factor of 2.

Section 4.2.3. Image Encoding Through Leveling of Edges.

A property of the level curve encoding scheme is that it preserves edges. This occurs because edges are at high contrast points of the images through which the level curves tend to pass. To capitalize on this observation, a similar scheme was developed which encodes an image using contours which run along edges in the image.

In this scheme the edges, or high contrast points, of an image are extracted and connected to form contours. The grey level at each edge contour is averaged for each side of the edge, and the whole contour is assigned two values: one for each side of the edge. The contours can be compactly encoded, as above, and the image is reconstructed by interpolating the grey levels between the edge pixels.

Due to time constraints, this scheme and the TEC scheme were not united, so the results reflect compression of the date using a lossless standard technique. From the experimentation with the TEC scheme, we hope to achieve an improvement in compression by roughly a factor of 2.

Section 4.3. Fractals and Wavelets: Theoretical Investigations.

Transform methods.

One efficient way of determining the local frequency content of the image is through transforms related to the Fourier transform. These, in order of increasing generality, are the Fourier transform, the Wigner transform, and wave packet transforms. The Wigner transform is given by the integral

$$W(f) = \int e^{-2\pi i \xi \mathbf{p}} f(\mathbf{x} + \frac{1}{2}\mathbf{p}) f(\mathbf{x} - \frac{1}{2}\mathbf{p}) d\mathbf{p},$$

and a generalized wave packet transform is given by the integral

$$P_\phi^f(\mathbf{p}, \mathbf{q}, t) = \int e^{-\pi \mathbf{p}\mathbf{q} + 2\pi \mathbf{p}\mathbf{x}} t^{1/2} \phi[t^{1/2}(\mathbf{x} - \mathbf{q})] f(\mathbf{x}) d\mathbf{p}$$

where $\phi(\mathbf{x})$ is a generalized Gaussian (see below). A little consideration of the definition of $W(f)(\mathbf{x}, \xi)$ will show that this transform gives the local frequency content of the function f in a neighborhood of x. The expression for $P_\phi^f(\mathbf{p}, \mathbf{q}, t)$ is

the inverse Fourier transform for $W(f)$ if we set $t = 1$ and $\phi = f$. Details may be found in Folland's 1989 monograph, pp56-63 and 142-169.

Wavelet transform methods.

We investigated relationships between fractal methods and wavelet transforms. The objective of this phase of the study was to look for ways to improve the choice of affine iterated function systems. One rationale for doing this is that our current image compression scheme does not preserve edges well. Because edges contribute heavily to human perception of image quality, it is important to amend this weakness. An appropriate wavelet transform yields a powerful tool to extract edge information from an image on a variety of scale lengths. Figure B.5 shows edges extracted from a wavelet transform of "Lena" at different scale lengths. Edges correspond to zero crossings of a wavelet transform of an image and the image may be reconstructed, by standard methods, from such zero crossings.

Edge information may be used in conjunction with fractal methods in various ways. One approach is to force the coding method to preserve edges by using a error measure which weights the edge error more heavily. (Recall that our encoding scheme chooses transforms on the basis of an integrated error.) An alternative approach is to regard the edges as one-dimensional fractal curves. The edges themselves may be coded as iterated transforms, using a 1-dimensional analog of the 2-dimensional method used for images, as has been detailed above.

Generalized Gaussian functions give one class of wavelet transforms. In order to clarify the relationship between affine fractals and this particular wavelet transform, we consider the behavior of the generalized Gaussian under affine transformations. A generalized Gaussian is given by a function:

$$\phi_{\mathbf{A},\mathbf{b},d}(\mathbf{x}) = \exp(\mathbf{x}^t\mathbf{A}\mathbf{x} + \mathbf{b}^t\mathbf{x} + d),$$

where $\mathbf{x}$ is a vector in the plane, $\mathbf{A}$ is a symmetric, negative definite matrix, and $\mathbf{b}$ is a fixed vector. There is also a complex-valued form, in which the entries of $\mathbf{A}$ and $\mathbf{b}$ are complex numbers and the real part of $\mathbf{A}$ is required to be negative definite. The set of generalized Gaussians is preserved under non-singular affine transformations and the orbits (up to a complex factor) are specified by the cogredience class of $\mathbf{A}$. According to a theorem of Sylvester, any two real symmetric matrices are cogredient if they have the same rank and the same signature (Jacobson, 1953). The closure property may be seen by a simple calculation. If $\mathbf{x} \mapsto \mathbf{B}\mathbf{x} + \mathbf{c}$ is an affine transform A, then $\phi(\mathbf{x})$ is transformed into

$$A(\phi_{\mathbf{A},\mathbf{b},d})(\mathbf{B}\mathbf{x} + \mathbf{c}) = \exp(\mathbf{x}^t\mathbf{B}^t\mathbf{A}\mathbf{B}\mathbf{x} + \mathbf{c}^t\mathbf{A}\mathbf{x} + \mathbf{x}^t\mathbf{A}\mathbf{c} + \mathbf{b}^t\mathbf{B}\mathbf{x} + \mathbf{c}^t\mathbf{A}\mathbf{c} + \mathbf{b}^t\mathbf{c} + d).$$

The latter function is of the form $\phi_{\mathbf{A}',\mathbf{b}',d'}$, where $\mathbf{A}' = \mathbf{B}^t\mathbf{A}\mathbf{B}, \mathbf{b} = 2\mathbf{c}^t\mathbf{A} + \mathbf{b}^t\mathbf{B}$ and $d' = \mathbf{c}^t\mathbf{A}\mathbf{c} + \mathbf{b}^t\mathbf{c} + d$. Because $\mathbf{A}$ is symmetric and negative definite the equation

$$2\mathbf{c}^t\mathbf{A} + \mathbf{b}^t\mathbf{B} = \mathbf{e}^t$$

has a unique solution $\mathbf{c}$. Therefore by proper choice of affine transforms, the quadratic part $\mathbf{x}^t\mathbf{A}\mathbf{x}$ is specified up to cogredience class. The linear part $\mathbf{b}^t\mathbf{x}$ is arbitrary, and we are left with a constant factor which cannot be specified.

Future work should clarify the relationship between image coding by wavelets and image coding by iterated function systems. The invariance of the set of generalized Gaussian functions under affine transformations indicates that the analog of a fractal should be a sum of Gaussians which is preserved under an iterated function system. In particular, if

$$f = \sum_i \phi_{\mathbf{A}_i,\mathbf{b}_i,d_i}$$

then we look for a system of affine transformations $\{A_1, A_2, \ldots, A_N\}$ with weights $\{w_1, w_2, \ldots, w_N\}$ such that

$$f = \sum_{i=1}^{N} w_i (A_i)^{-1}(f) = \sum_i \phi_{\mathbf{A}_i',\mathbf{b}_i',d_i'}.$$

Because we require

$$\sum_i \phi_{\mathbf{A}_i',\mathbf{b}_i',d_i'} = \sum_i \phi_{\mathbf{A}_i,\mathbf{b}_i,d_i}$$

the summations need not be unique.

Our survey of the literature has revealed some other connections between generalized harmonic analysis and affine transformations, in particular through representations of the extended metaplectic representation and the inhomogeneous symplectic group. The metaplectic representation is a representation of the symplectic group (the group of $2n \times 2n$ matrices which preserve the symplectic form $[(\mathbf{p},\mathbf{q}),(\mathbf{p}',\mathbf{q}')] = \mathbf{p}\mathbf{q}' - \mathbf{p}'\mathbf{q}$, on vectors with $2n$ components. The symplectic group is generated by matrices

$$\begin{pmatrix} A & 0 \\ 0 & \mathbf{A}^{*-1} \end{pmatrix}, \begin{pmatrix} \mathbf{I} & 0 \\ \mathbf{C} & \mathbf{I} \end{pmatrix}, \text{ and } \begin{pmatrix} 0 & \mathbf{I} \\ -\mathbf{I} & 0 \end{pmatrix}$$

The action is given, up to sign, by

$$m\left[\begin{pmatrix} \mathbf{A} & 0 \\ 0 & \mathbf{A}^{*-1} \end{pmatrix}\right] f(\mathbf{x}) = \frac{1}{\sqrt{\det(\mathbf{A})}} f(\mathbf{A}^{-1}\mathbf{x})$$

$$m\left[\begin{pmatrix} \mathbf{I} & 0 \\ \mathbf{C} & \mathbf{I} \end{pmatrix}\right] f(\mathbf{x}) = \pm e^{-\pi \mathbf{C}\mathbf{x}} f(\mathbf{x})$$

$$m\left[\begin{pmatrix} 0 & \mathbf{I} \\ -\mathbf{I} & 0 \end{pmatrix}\right] f(\mathbf{x}) = i^{-n/2} F^{-1} f(\mathbf{x}),$$

where F is the Fourier transform. The extended metaplectic representation is a representation of the semidirect product of the Heisenberg and symplectic groups. The semidirect product is given as pairs of operators (X, A), where X is in the

Heisenberg group and A is in the symplectic group. The group product is given by $(X, A)(X', A') = (X(AX'), AA')$, and the representation is given by $w(X, A) = r(X)m(A)$. The Heisenberg group acts as follows for $X = (\mathbf{p}, \mathbf{q}, t)$:

$$Xf(\mathbf{x}) = e^{2\pi t} e^{2\pi \mathbf{q}\mathbf{x} + \pi \mathbf{p}\mathbf{q}} f(\mathbf{x} + \mathbf{p}).$$

Because the product in the Heisenberg group is given by

$$(\mathbf{p}, \mathbf{q}, t)(\mathbf{p}', \mathbf{q}', t') = (\mathbf{p} + \mathbf{p}', \mathbf{q} + \mathbf{q}', t + t' + \frac{1}{2}(\mathbf{p}\mathbf{q}' - \mathbf{q}\mathbf{p}')),$$

it can be seen that the extended metaplectic representation contains the usual action of the affine group.

The inhomogeneous symplectic group is another extension of the symplectic group, which has a more obvious relationship to the affine group. In this case the semidirect product is the product of R^{2n} with the symplectic group, and the group multiplication is

$$(w, A)(w', A') = (w + Aw', AA').$$

The representation, as before, is given by $w(X, A) = r(X)m(A)$. This is a projective representation, and the extension of this representation by a representation of the circle group is the extended metaplectic representation.

Other connections between generalized Gaussians and affine transformations.

It is easy to show that the set of generalized Gaussians is preserved under the convolution product. This may be derived from results on the oscillator semigroup (Folland, page 231) or calculated directly. Furthermore, affine transformations preserve the convolution product, up to a constant factor. If we write

$$\gamma_{[\mathbf{A},\mathbf{x_0},c]} = \exp[(\mathbf{x} - \mathbf{x_0})^t \mathbf{A}(\mathbf{x} - \mathbf{x_0}) + c]$$

where, as before the real part of $\mathbf{A}$ is negative definite, then

$$M\gamma_{[\mathbf{A},\mathbf{x_0},c]} = \gamma_{[\mathbf{M}^t\mathbf{A}\mathbf{M},\mathbf{M}^{-1}(\mathbf{x_0}-\mathbf{b}),c]},$$

where $M(x) = \mathbf{M}(\mathbf{x}) + \mathbf{b}$, and

$$\gamma_{[\mathbf{A},\mathbf{x_0},c]} * \gamma_{[\mathbf{B},\mathbf{y_0},c]} = \gamma_{[\mathbf{A}^t(\mathbf{A}+\mathbf{B})^{-1}\mathbf{A}+\mathbf{A},\mathbf{x_0}+\mathbf{y_0},c']}$$

where $c' = c - \frac{1}{2}\log(\det(\mathbf{A} + \mathbf{B}))$ and $*$ represents the convolution product.

This may be useful in shortening the search for affine transformations which map one part of the image into another. In particular, we may be able to use a wavelet decomposition and the expression for the convolution to comute the best correlation between regions of the image.

Affine transformations and theta functions.

One definition for the theta function is given via a generalization of the Fourier transform. We let ${}_1x$ be a vector in R^k and ${}_jx$ stand for a vector in the space jR^k of homogeneous polynomials of degree j in R^k. There is an obvious inner product in this space, so we can define the nonlinear Fourier transform

$$F(\tau^1,\ldots,\tau^J) = \int f({}_1\mathbf{x})\exp[i\tau^1 \cdot {}_1\mathbf{x},\ldots i\tau^J \cdot {}_J\mathbf{x}]d\mathbf{x}.$$

If we take f to be $\sum \delta_1\mathbf{n}$, where ${}_1\mathbf{n}$ is the lattice of vectors with integer coordinates in R^k and $J=2$, then we get the function $\theta_3(\mathbf{x},\mathbf{z})$. Furthermore,

$$\int f(M_1\mathbf{x})\exp[i\tau^1 \cdot {}_1\mathbf{x},\ldots,i\tau^J \cdot {}_J\mathbf{x}]d\mathbf{x}$$

$$= \frac{1}{det(\mathrm{M})}\int f({}_1\mathbf{x})\exp[i\tau^1 \cdot M^{-1}\,{}_1\mathbf{x},\ldots,i_J(M^{-1})_J\mathbf{x}]d\mathbf{x}$$

$$= \frac{1}{det(\mathrm{M})}\int f({}_1\mathbf{x})\exp\{i[M^{-1}]*\tau^1 \cdot\ {}_1\mathbf{x},\ldots,i[{}_J(M^{-1})]*\tau^J \cdot {}_J\mathbf{x}\}d\mathbf{x}$$

$$= \frac{1}{det(\mathrm{M})}F([M^{-1}]*\tau^1,\ldots,[{}_J(M^{-1})]*\tau^J),$$

where $*$ denotes the adjoint and the pre-subscript denotes the appropriate symmetric product. This shows that approximate symmetries under the affine group of the function f can be represented as approximate symmetries of the affine group acting on the nonlinear Fourier transform of f. This indicates that searching zero-crossings should be a good way to find appropriate transforms to code an image.

Section 5.0. Comparison With Other Image Compression Schemes.

This section contains some results from other image compression schemes. This section is by no means complete; the omission or inclusion of any results is happenstance. The following graph shows results from a collection of recent publications, listed below.

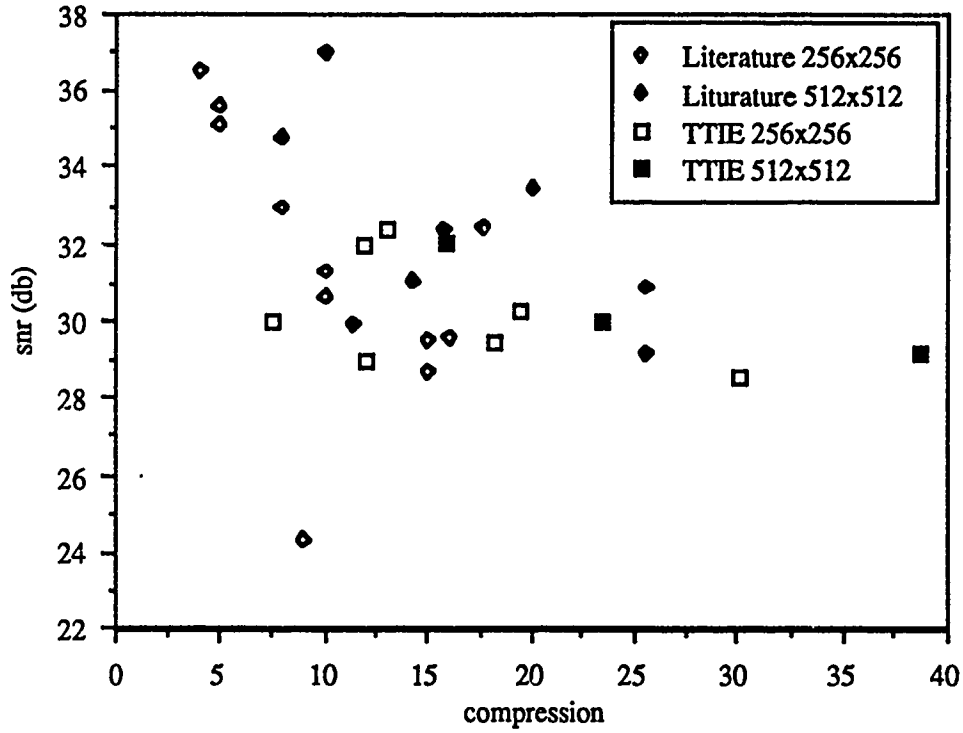

Figure 4. Signal to noise ratio vs. Compression from a variety of recent publications and from the TTIE scheme.

The figure is somewhat misleading in that not all of the images compressed are the same, or even the same size. The points are segragated into image size, results from the literature and our results. The high fidelity, low compression results come from a paper which used particularly simple images to encode. Nevertheless, the TTIE scheme compares well with other results.

Table 3 tabulates the results in the image and gives the references to the gathered data. The references for the gathered data follow:

[1] Analysis/Synthesis for Subband Image Coding, Smith, Eddins, IEEE Trans. Speech Acoustics and Signal Processing, Vol. 38, No. 8, Aug 1990.

[2] Adaptive Cosine Transform Coding of Images, Ngan, Leong, Singh, IEEE Trans. Speech Acoustics and Signal Processing, Vol. 37, No. 11, Nov 1989.

[3] Image Coding - From Waveforms to Animation, R. Forchheimer, T. Kronander, IEEE Trans. Speech Acoustics and Signal Processing, Vol. 37, No. 12, Nov 1989.

[4] Subband Image Coding, Woods and Oneil, IEEE Trans. Speech Acoustics and Signal Processing, Vol. 34, No. 10, Oct 1986.

[5] Non Linear Space-Carient Postprocessing of Block Coded Images, Rmammuthi, Gersho, IEEE Trans. Speech Acoustics and Signal Processing, Vol. 34, No. 5, May 1986.

[6] Sliding Block Entropy Encoding of Images, Cohen and Woods, M7.1 IEEE, 1989.

[7] Pruned Tree-Structured Vector Quantization in Image Coding, Riskin, Daly and Gray, M7.2 IEEE, 1989.

Reference	Compression	SNR (db)	Image Size	Image
1	10	30.7	256	simple
1	10	31.3	256	simple
1	5	35.1	256	simple
1	5	35.6	256	simple
1	15	28.7	256	simple
1	15	29.5	256	simple
2	20	33.5	512	Lena
2	10	37.0	512	Lena
3	9	24.3	256	Lena
4	16	29.6	?	?
4	8	33.0	?	?
4	4	36.5	?	?
5	11.4	29.9	512	Lena
6	8	34.8	512	Lena
6	17.7	32.5	512	Lena
6	14.3	31.1	512	Lena
7	25.6	29.2	512	Lena
7	25.6	30.92	512	Lena
7	15.68	32.43	512	Lena
TTIE	7.5	30.0	256	city
TTIE	19.54	30.28	256	collie
TTIE	11.8	31.96	256	collie
TTIE	8.1	28.48	256	collie
TTIE	18.3	29.4	256	colie
TTIE	13.1	32.4	256	girl
TTIE	12	28.9	256	Lena
TTIE	38.7	29.2	512	Lena
TTIE	15.91	32.1	512	Lena
TTIE	23.5	30.0	512	Lena

Table 3

Section 6.0. Conclusion.

The tiled transform image encoding scheme yields good results which are comparable to the latest results attainable by other schemes. It is sufficiently mature to be implemented in hardware, and this should be one of the next goals of pursuing "fractal" image encoding further. The research into the subject, however, should not be considered complete. The alternative schemes presented in the paper should

be followed to see if they yield even better results. Given that the TTIE scheme was only explored for a short time, there is good reason to expect that further research will lead to such results.

Finally, the mathematical modeling of the underlying processes should be extended. As we demonstrated, the current level of understanding of the scheme is rather incomplete. Basing a scheme on "motivational" arguments cannot be expected to provide optimal results. A thorough research effort into building a good model of the process should lead to much better results.

Section 7.0. Recommendations.

We recommend further research and development along several lines. First, the tiling scheme is sufficiently mature to pursue hardware implementation for possible real time applications. Second, we expect that the application of fractal techniques with the other image processing methods presented, such as contour detection, will lead to even better algorithms. This avenue of research should be developed. Finally, research into the mathematical foundations of the subject has barely begun. We believe that a program which integrates hardware engineering, software development and mathematical research will yield the best results in the long run.

More specific recommendations will be found in our followup Phase II proposal.

Acknowledgment

In preparing this six month Phase I research program, Mr. Dan Greenwood served as Principal Investigator. Dr. Yuval Fisher developed many of the algorithms and software. NETROLOGIC's consultant, Dr. Albert Lawrence, explored the mathematical foundations and suggested many novel approaches in the research. Spencer Menlove wrote many excellent programs and made many helpful suggestions to improve the later algorithms.

Appendix A: Other Work

In this appendix we describe other work carried on during the phase I research which is of secondary interest. The section is organized into small subsections which briefly describe the work and results.

Lossless Compression.

While the compression ratios attainable with lossless compression schemes are far lower than those obtained with lossy schemes, the utility of lossless schemes is higher for many applications. We investigated briefly the idea of compressing the difference between an original image and a highly compressed version of the image. The results were encouraging, though not spectacular. Compression of our standard test images ranged from 1.3 to 1.9. Further work along these lines is warranted, in view of the importance of lossless compression schemes.

Fourier Methods.

We investigated the use of FFT methods in conjunction with the TTIE scheme. The results were uniformly poor. The frequency domain does not display the type of coherency or adjacent pixel correlation which the TTIE scheme can capitalize on in order to achieve good compressions.

Alternative Tile Classification Schemes.

Using alternative classification schemes for tiles is somewhat technical. Reference [1] has further details. The domain tiles used to encode an image are classified in order to speed the search needed to find a "good" domain tile. Using a classification scheme generally results in poorer fidelity, because there is no guarantee that the optimal domain will be found in the class searched. We investigated several methods based on correlation methods and moments. The results are not definitive, being better in some features and worse in others as compared with the present scheme.

Elimination of Compression Artifacts.

One weakness of the current compression scheme is the appearance of artifacts. In order to eliminate these we attempted to postprocess the image by smoothing along the boundaries of the range tiles. This was successful, often resulting in an even lower rms error. Of the attempts to remove compression artifacts, this was the most successful. This type of postprocessing can, and should, be utilized in any final implementation of TTIE scheme. While the TTIE scheme is still in development, we felt it was better to concentrate on optimizing the results obtainable with the method, rather than attempt to do a good job cleaning up afterwards.

Image Preprocessing.

In an effort to improve the ability of the encoding scheme in encoding images, we investigated two image preprocessing methods. The first, based on Wigner transforms was disappointing. This transform lead to data which was complicated. The second method using a wavelet transform, is potentially useful. We used this

transform as a method of edge detection, though we had hoped to find a deeper relationship between it and the affine transformations used to encode images (see section 4.3). This research did not lead to any concrete results.

Alternative Tiling Methods.

We attempted to improve the fidelity of a given tiling by using linear combinations of domain tiles. Initial results were very encouraging, having greatly reduced artifacts. Using more than one domain tile significantly reduced the encoding error, but decreased the overall compression. Without careful classification, the search time needed to find even a moderate approximation to an optimal fit is prohibitive. Our last approach to using several domain tiles in an encoding was to encode a range optimally and then encode the resulting error. Due to time constraints, we did not carefully study the relationship between the encoding fidelity and the compression ratio as compared with the TTIE scheme. Such a study is warranted.

Appendix B: Images

References

[1] Iterated Transform Image Compression, Yuval Fisher, R.D. Boss, E.W. Jacobs, submitted to IEEE Trans. Speech, Acoustics, and Signal Processing.

[2] Barnsley, M. Fractals Everywhere. Academic Press. San Diego, 1989.

[3] Michael F. Barnsley, Alan D. Sloan, *A Better Way to Compress Images.* Byte, January 1988.

[4] E.W. Jacobs, R.D. Boss ,Y. Fisher, Fractal-Based Image Compression II, Naval Ocean Systems Center Technical Report 1362, June 1990.

[5] John E. Hutchinson, *Fractals and Self Similarity.* Indiana University Mathamatics Journal, Vol. 35, No. 5. 1981.

[6] Stephen Demko, Laurie Hodges, Bruce Naylor, *Construction of Fractal Objects with Iterated Function systems.* ACM Vol. 19, No. 3. 1988.

[7] Jacquin, A., A Fractal Theory of Iterated Markov Operators with Applications to Digital Image Coding. Doctoral Thesis. Georgia Institute of Technology.1989.

[8] W. Rudin, Real and Complex Analysis. McGraw Hill, New York. 1974.

Figure B.1

The decoding process starting with an initial image containing a pattern. The images show one, two and ten application of all of the encoding transformations. The lower right image is reconstructed from a 10.0:1 compression of figure B.2 at a signal to noise ratio of 29.5db.

Figure B.2

The original Lena at size 256x256.

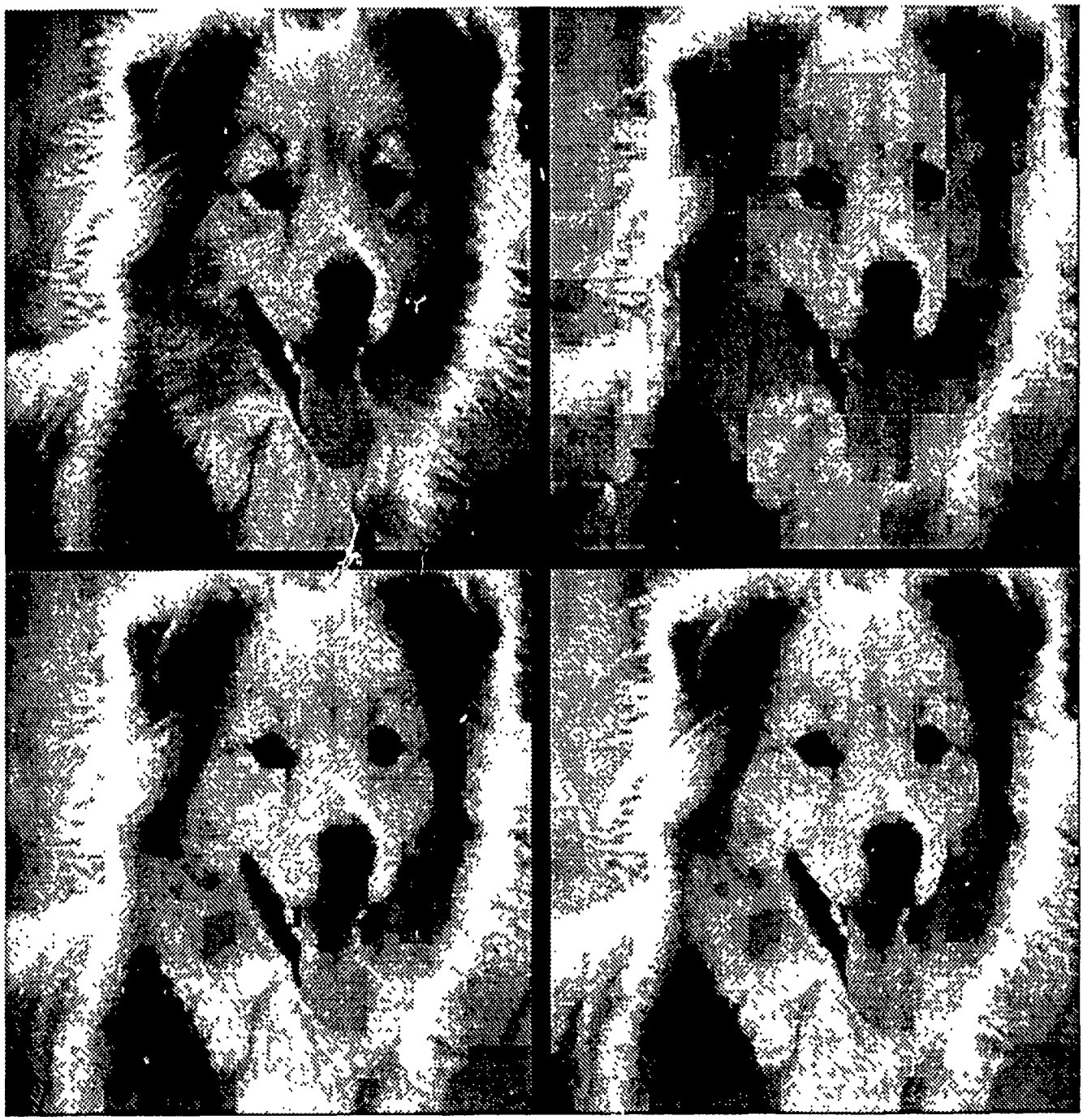

Figure B.3

The original collie at size 256x256 8bpp (upper left). Collie compressed 63.0:1 with a signal to noise ratio of 25.2db (upper right). Collie compressed by a factor of 28.2 with a signal to noise ratio of 29.3db (lower left). Collie compressed at 16.6:1 at 30.1db snr (lower right).

Figure B.4

A reconstruction of a 512x512 original version of Lena compressed at 38.7:1 with a signal to noise ratio of 29.2db.

Figure B.5

A wavelet transform of a 256 x 256 version of Lena resulting in the extraction of edge information.

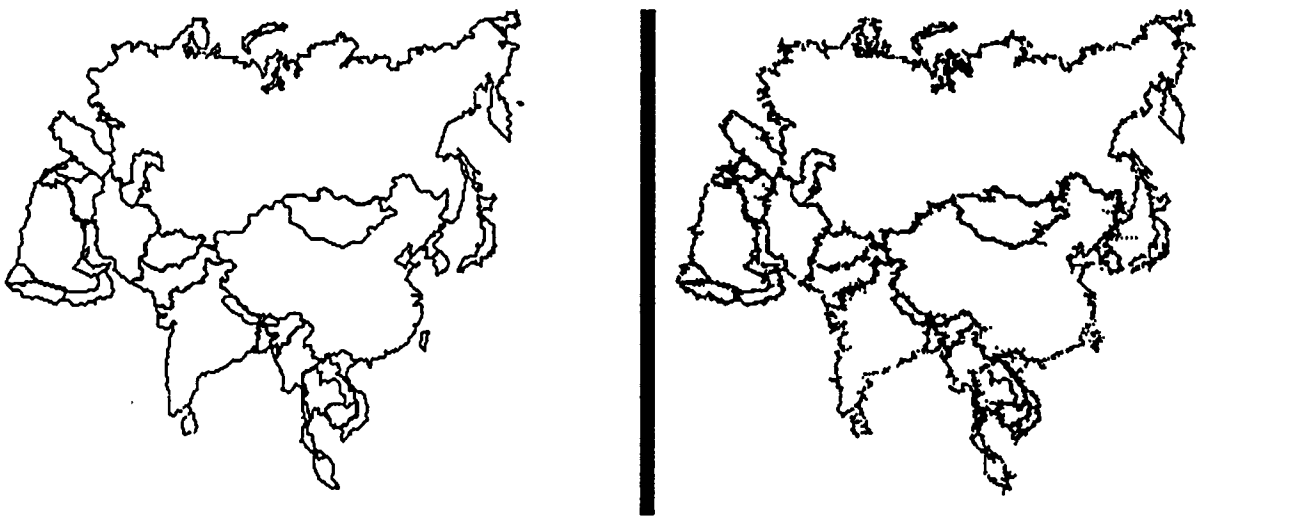

Figure B.6

An original map of Asia (left) and a fractally encoding of that map. The resulting compression depends on the storage method of the original map. For very efficient schemes, the compression is still near 1 at this preliminary stage of development of the algorithm.

Figure B.7

Contour encoding of images. Lena encoded with 9 (upper) and 15 (lower) contours with respective compressions of 5.9 and 4.2 and signal to noise ratios of 26.3db and 26.5db. We expect to increase the compression once the contours are encoded as transformations. The contours are shown on the right of each image.

Figure B.8

Edge contour encoding of images. Lena encoded with two different edge sensitivities. (20 upper and 10 lower). The respective compressions are 5.7 and 3.0 with signal to noise ratios of 22.5db and 27.4db. The signal to noise ratio does not reflect the true image fidelity, since the edges are very well preserved by this scheme. We expect to increase the compression once the edge contours are encoded as transformations. The edge contours are shown on the right of each image.

Figure B.9

A modified Edge contour encoding of images. Lena encoded with two different edge sensitivities. (20 upper and 10 lower). The respective compressions are 7.2 and 3.5 with signal to noise ratios of 21.6db and 24.4db. The signal to noise ratio does not reflect the true image fidelity, since the edges are very well preserved by this scheme. We expect to increase the compression once the edge contours are encoded as transformations. The edge contours are shown on the right of each image.