SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

REPORT DOCUMENTATION PAGE

	1b. RESTRICTIVE MARKINGS
AD-A222 203 E	3. DISTRIBUTION / AVAILABILITY OF REPORT Unclassified/Unlimited
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S) AFOSR-TR- 90-0581
6a. NAME OF PERFORMING ORGANIZATION Department of Mathematics Carnegie Mellon University	6b. OFFICE SYMBOL (If applicable)
7a. NAME OF MONITORING ORGANIZATION Air Force Office of Scientific Research	
6c. ADDRESS (City, State, and ZIP Code) Pittsburgh, PA 15213	7b. ADDRESS (City, State, and ZIP Code) Bolling Air Force Base, D.C. 20332-6448 (BG 410)
8a. NAME OF FUNDING / SPONSORING ORGANIZATION AFOSR	8b. OFFICE SYMBOL (If applicable) AFOSR/NM
9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER AFOSR-87-0137	
8c. ADDRESS (City, State, and ZIP Code) Bldg. 410 Bolling AFB, DC 20332-6448	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. 61102F; PROJECT NO. 2304; TASK NO. A9; WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)

Radon Transform Analysis of a Probabilistic Method for Image Generation

12. PERSONAL AUTHOR(S)

Berger, Marc A.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final	FROM 87/4/1 TO 90/3/31	1990 March 30	

16. SUPPLEMENTARY NOTATION

(Square) (Cubed)

MAY 30 1990

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

During the tenure of this research grant the principal investigator ~~(PI)~~ worked in the areas of random iterated function systems (IFS) and computer image generation. Random IFS afford a powerful new technique for generating images. The image is constructed as the attractor for a random dynamical system in $\mathbf{R}^2$ or $\mathbf{R}^3$, typically a Markov process or some spin-off of such a process. The image generation algorithm proceeds by simply generating the appropriate random dynamical system (via a random number generator) and plotting the points along a single orbit. The random dynamics evolve through compositions of independent identically distributed (i.i.d.) affine transformations.

The images obtained in this way include fractals, landscape, natural scenes, simple geometrical shapes, smooth curves and surfaces, rough curves and surfaces, wavelets, B-splines, Beziér curves and interpolants. In fact this technique produces one of the most efficient and parallelizable algorithms for high-speed curve and surface generation.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	(Unclassified)
22a. NAME OF RESPONSIBLE INDIVIDUAL Dr. Arje Nachman	22b. TELEPHONE (Include Area Code) 202-767-4939
	22c. OFFICE SYMBOL AFOSR/NM

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted. All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

90 05 25 ~~186~~ 186

RADON TRANSFORM ANALYSIS OF A PROBABILISTIC METHOD FOR IMAGE GENERATION

Marc A. Berger
School of Mathematics
Georgia Institute of Technology
Atlanta, GA 30332

31 March 1990

Final Technical Report for Period 1 April 1987 - 31 March 1990

Grant No. AFOSR-87-0137

Prepared for

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH
Bolling Air Force Base, DC 20332-6448

Table of Contents

Summary

During the tenure of this research grant the principal investigator (PI) worked in the areas of random iterated function systems (IFS) and computer image generation. Random IFS afford a powerful new technique for generating images. The image is constructed as the attractor for a random dynamical system in $\mathbf{R}^2$ or $\mathbf{R}^3$, typically a Markov process or some spin-off of such a process. The image generation algorithm proceeds by simply generating the appropriate random dynamical system (via a random number generator) and plotting the points along a single orbit. The random dynamics evolve through compositions of independent identically distributed (i.i.d.) affine transformations.

The images obtained in this way include fractals, landscape, natural scenes, simple geometrical shapes, smooth curves and surfaces, rough curves and surfaces, wavelets, B-splines, Beziér curves and interpolants. In fact this technique produces one of the most efficient and parallelizable algorithms for high-speed curve and surface generation.

Key questions are how to find the appropriate random dynamics to generate a given target image (the "encoding problem"), and how to adapt the algorithm so as to produce different classes of images. There were several research accomplishments in these directions under this grant. They are listed here, and elaborated on below in the sections which follow.

I) Encoding via the Collage Property:

Together with J.-P. Vidal (Ph.D. student, Carnegie Mellon University) the PI developed an automated encoding scheme. This encoder works by sequentially collaging a given polygonalized target image by affine copies of itself. It essentially solves a puzzle by moving and adjusting the pieces locally in an iterative fashion, so that they efficiently collage the target. (This is not quite the same as a puzzle – since the pieces are allowed to overlap here.)

II) Encoding via Convexity:

Together with J.-P. Vidal the PI discovered an important property satisfied by the extreme points of (the convex hull of) the attractor for an IFS. Namely, every extreme point is the image of some extreme point under one of the affine maps. Equivalently, the set of extreme points is invariant under the inverse dynamics. Thus by examining the extreme points of the attractor, one can extract information about the affine maps which generate it. This extreme point phenomena was observed by the PI to hold as well for more general iterated function systems than affine – for example Julia sets for iterates of complex polynomials.

III) Encoding via Discrete Optimization:

Together with M. Perrugia (Ph.D. student, Carnegie Mellon University) the PI studied the discrete Markov chain obtained by truncating the continuous process in $\mathbf{R}^2$ to the centers of the pixels in which it lands. It was shown that as pixel dimension decreases to zero, stationary distributions for the discrete truncated chains converge weakly to that of the continuous IFS process. This then enables one to approach the encoding problem through discrete optimization.

IV) Encoding via the Radon Transform:

The Radon transform of the invariant measure for an IFS satisfies a difference equation, with difference parameters related to the affine maps of the IFS. Knowing the target image as a color or grey-level image amounts to knowing the invariant measure, and from this one can obtain its Radon transform via plane wave scans. So for this approach encoding becomes the inverse problem of recovering the parameters of a difference equation from measurements of its solution (the Radon transform).

V) Mixing Images:

Together with M. Barnsley (Iterated Systems, Inc.) and M. Soner (Carnegie Mellon University), the PI developed a framework for mixing image textures. For example one could start with a given leaf texture and tree texture, and then create a tree with those leaves growing on its branches. The mixing involves constructing a stochastic process which jumps between various Markov processes, transferring states from one to the other via a graph structure. The nodes of this graph represent simple (un-mixed) IFS processes, and the edges represent crossovers, whereby a point in the orbit of one process gets replaced by the corresponding point in the orbit of an adjacent process.

VI) Real-Time Animation and Vectorization:

IFS images can be animated by continuously varying the IFS parameters (the coefficients of the affine maps and the probabilities assigned to them). The IFS algorithm is highly parallel, and animation is also parallel since there are no dependencies between one image and the next. (That is, they can all be generated simultaneously.) The PI used the vectorization and multi-processor capabilities of the CRAY Y-MP/832 at the Pittsburgh Supercomputing Center to optimize IFS animation, thereby generating the frames in real-time (30 frames/sec.). This work also demonstrates the continuous dependence of the image on the IFS parameters, as continuous variation of parameters

leads to a continuous flow of images.

VII) Wavelets and Curve Generation

Most recently the PI discovered how to set up a random IFS to generate wavelets and various curves arising from sub-division methods. These include B-splines, Bezier curves and line averaging interpolants. In fact for a whole variety of computer image generation algorithms involving recursive tree traversal, the PI has shown how one can construct a simple random process which generates the same image. The image then evolves as the orbit of a single trajectory. This involves products of i.i.d. row stochastic matrices, as described below.

Introduction

The basic IFS image generation algorithm is illustrated in Figure 1. The leaf is generated as follows. Pick any point $X_0 \in \mathbf{R}^2$. There are four affine transformations T: $x \mapsto Ax + b$ listed on the top of this Fig., and four probabilities p_i underneath them. Choose one of the transformations at random, according to the probabilities p_i – say T_k is chosen, and apply it to X_0, thereby obtaining $X_1 = T_k X_0$. Then choose a transformation again at random, independent of the previous choice, and apply it to X_1, thereby obtaining X_2. Continue in this fashion, and plot the orbit $\{X_n\}$. The result is the leaf shown. By tabulating the frequencies with which the points X_n fall into the various pixels of the graphics window, one can actually plot the empirical distribution $\frac{1}{n+1} \sum_{k=0}^{n} \delta_{X_k}$, using a grey scale to convert statistical frequency to grey level. The darker portions of the leaf correspond to high probability density.

I. Encoding via the Collage Property

Given a target set C, the affine mappings T_i for the IFS to (nearly) reproduce it are obtained through the following optimization problem

$$\underset{\text{affine contractions } T_i}{\text{MAXIMIZE}} \left[\text{area}(T_i C \cap R_{i-1}) - \lambda \, \text{area}^2(T_i C \backslash C)\right]$$

where R_i are the residual sets $R_0 = C$,

$$R_i = R_{i-1} \backslash T_i C$$

and λ is an adjustable parameter. This optimization problem is solved sequentially for $T_1, T_2, \ldots$. The motivation behind this technique, and an adaptive algorithm developed by J.-P. Vidal for solving this problem were described in my Second Annual Report (dated 12 April 1989), and in my Research Progress and Forecast Report for the First Additional Period of Research (dated 14 October 1988).

II. Encoding via Convexity

The PI has shown that the extreme points of the attractor C for an IFS are invariant under the inverse dynamics. That is, every extreme point is the image of some extreme point under one of the affine mappings. Equivalently

$$E \subseteq \bigcup_i T_i E \tag{1}$$

where E is the set of extreme points of (the convex hull of) C, and T_i are the affine maps for an IFS which generates C. This is illustrated in Figure 2. Since an affine map $T: \mathbf{R}^2 \rightarrow \mathbf{R}^2$ is completely determined by its action on three points, this condition (1) can be used to partially or fully determine the maps T_i from knowledge of the image C. In fact condition (1) was observed by the PI to hold in more general settings of Julia sets, where the maps T_i are the inverse branches of a complex polynomial f(z). In this case $f(E) \subseteq E$, where E is the set of extreme points of the Julia set.

J.-P. Vidal implemented an algorithm for finding all possible sets of affine maps T_i which are consistent with (1). This was described in my First Annual Technical Report (dated 15 April 1988), and in my Research Progress and Forecast Report for the First Additional Period of Research (dated 14 October 1988). It was published in [5]. (See the list of publications at the end of this report.)

III. Encoding via Discrete Optimization

The PI, together with M. Perrugia, studied a discrete IFS model, whereby a continuous IFS process is generated, and every point along the orbit is sequentially replaced with the center of the pixel in which it lies. It was shown that if the continuous IFS is strictly contractive, then as pixel size $\rightarrow 0$ any sequence of stationary distributions for the discrete IFS converges weakly to the (unique) stationary distribution for the continuous IFS. The discrete IFS need not have a unique stationary distribution, and conditions ensuring ergodicity of the discrete IFS when pixel size is sufficiently small are the subject of M. Perrugia's Ph.D. thesis.

The discrete IFS leads to a discrete formulation for the encoding problem. Given the

stationary distribution π satisfying $\pi P = \pi$, where P is a transition probability matrix having precisely N non-zero entries in each row, recover P. Typically the dimensions of P (say $(1024)^2 \times (1024)^2$ for a high resolution screen) are much larger than N (which is on the order of 10). There is in fact more structure on account of the affine-ness, so that the non-zero entries of P are constrained to form generalized diagonals. This was described in my Second Annual Technical Report (dated 12 April 1989).

IV. Encoding via the Radon Transform

Given a random variable $X \in \mathbf{R}^2$, the Radon transform $R(\lambda,z)$ is defined by

$$R(\lambda,z) = \mathbf{P}(X \cdot \lambda \leq z),$$

where $\lambda \in \mathbf{R}^2$ is a unit vector and $z \in \mathbf{R}$. If X has the stationary distribution for an IFS with affine maps $T_i: x \mapsto A_i x + b_i$ and probability weights p_i, then the invariance condition can be expressed as

$$R(\lambda,z) = \sum_i p_i R(\lambda_i,z_i) \tag{2}$$

where

$$\lambda_i = \frac{A_i^t \lambda}{||A_i^t \lambda||}, \quad z_i = \frac{z - b_i \cdot \lambda}{||A_i^t \lambda||}.$$

(If $A_i^t \lambda = 0$ then define

$$R(\lambda_i,z_i) = \chi_{\{b_i \cdot \lambda \leq z\}} .)$$

The PI has been developing an algorithm for recovering the maps T_i and weights p_i via (2), through knowledge of $R(\lambda,z)$. It proceeds by finding eigenvectors of the matrix parts A_i, through invariant half-planes. In fact this work was what led to discovery of the extreme point invariance condition described above in §II. Indeed one can use (2) to analyze supporting hyperplanes $X \cdot \lambda' \leq z'$ of the image, for which (λ',z') will be on the boundary of the support of $R(\lambda,z)$.

There is a similar difference equation for the Fourier transform (i.e., characteristic function)

$$\varphi(u) = \mathbf{E}e^{iX \cdot u}, \quad u \in \mathbf{R}^2.$$

Namely,

$$\varphi(u) = \sum_i p_i e^{b_i \cdot u} \varphi(A_i^t u). \tag{3}$$

The PI, together with R. Shenk (GA Tech.), has also been developing an algorithm for solving the encoding problem based on (3).

V. Mixing Images

Mixing theory involves an M×M transition probability matrix $P = (p_{ij})$ and M^2 (Borel) probability measures $\mu_{ij} \in \mathcal{P}(G)$, where G is the semi-group of affine transformations $\mathbf{R}^2 \to \mathbf{R}^2$. One imagines that M computer screens are each generating a stochastic process $(X_n(i): n \geq 0)$, for $1 \leq i \leq M$. (That is, i indicates screen number.) Let $(I_n(i): n \geq 1)$ be an i.i.d. sequence of "switching variables" distributed like the rows of P; that is,

$$\mathbf{P}(I_n(i) = j) = p_{ij}\,.$$

The mixed process then evolves as

$$X_{n+1}(i) = g_{n+1}(i,j)X_n(j), \quad \text{where } j = I_{n+1}(i).$$

Here $(g_n(i,j): n \geq 1)$ is an i.i.d. sequence distributed like μ_{ij}. Thus if one represents the screens as nodes of a directed graph, with processes evolving in them, then the $(n+1)^{st}$ point generated in node i is $g_{n+1}(i,j)$ applied to the n^{th} point from node j. The index $I_{n+1}(i)$ represents the crossover from node j to node i, along edge (j,i). Figure 3 illustrates a typical mixing scenario. This structure generates the coupling necessary to mix the individual images that would have been generated in each screen by the maps $(g_n(i,i): n \geq 1)$ alone. The details of this work appear in the Second Annual Technical Report (dated 12 April 1989), the Research Progress and Forecast Report for the First Additional Period of Research (dated 14 October 1988), the First Annual Technical Report (dated 15 April 1988) and the Research Progress and Forecast Report (dated 11 Sept. 1987). It was published in [1], [3], [5].

VI: Animation and Vectorization

Suppose one has two IFS, each corresponding to a specific image. One can set up a continuous flow of images from one to the other by interpolating the IFS parameters. What makes IFS animation

even more special is the ease with which one can rotate, scale, change perspective or vantage point, zoom in and out, or perform any affine transformation on an image. This is because the transformations can be built directly into the IFS.

IFS animation is highly parallel. The images for the various frames can all be generated simultaneously, since there are no dependencies among the different frames. Furthermore the same sequence of random numbers can be used for all the images. The PI was able to exploit the vector and multi-processor capabilities of the CRAY Y-MP/832 at the Pittsburgh Supercomputing Center to generate real-time animation. This was described in the Second Annual Technical Report (dated 12 April 1989), and a demo video tape was sent in to the Program Director. It was published in [2].

VII: Wavelets and Curve Generation

Many algorithms for generating computer images today involve a recursive tree-traversal. These include wavelets and solutions to dilation equations, sub-division refinement methods for generating B-splines and Beziér curves and line averaging methods for interpolants. Using ergodic theory the PI is able to produce random algorithms, in a very general setting, which generate the same images as the recursive ones. These images become attractors of random dynamical systems, and evolve simply as the trajectory of a *single* orbit. The random algorithms are very fast, involving only affine arithmetic, and are efficiently and highly parallelizable.

In sub-division and line averaging methods for curve generation one is given m×m row stochastic matrices P(0), ..., P(N−1) (i.e., all row sums are one − negative entries allowed) and a set of "control points" $v_1, \ldots, v_m \in \mathbf{R}^d$. These points form the vertices of a "control polytope" $C \subseteq \mathbf{R}^d$, which one can associate with the m×d matrix

$$\begin{bmatrix} v_1^t \\ \cdots \\ v_m^t \end{bmatrix}$$

(This correspondence is well-defined if one thinks of C as an "ordered" polytope.) If $P = (p_{ij})$ is an m×m row stochastic matrix then one can identify an action of P on C; namely, PC. Equivalently PC is the polytope with vertices $v_1', \ldots, v_m'$ where

$$v_i' = \sum_j p_{ij} v_j' .$$

Under suitable conditions on the P(i)'s it can be shown that

$$P(i_n) \cdots P(i_1)C$$

converges to a singleton as $n \to \infty$, for any sequence $(i_1, i_2, \ldots)$. By choosing the P(i)'s appropriately one generates a smooth curve $\mathcal{A}$ (the "attractor") as

$$\mathcal{A} = \cup(\lim_n P(i_n) \cdots P(i_1)C\text{: all sequences } (i_1, i_2, \ldots))$$

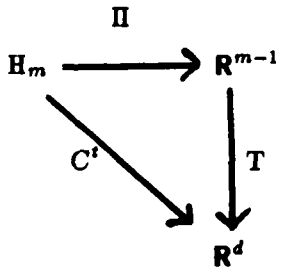
Let $H_m \subseteq \mathbf{R}^m$ be the hyper-plane

$$H_m = \{x \in \mathbf{R}^m\text{: } \sum_i x_i = 1\},$$

and let Π denote the projection $\mathbf{R}^m \to \mathbf{R}^{m-1}$ which lops off the last component

$$\Pi\text{:} \begin{bmatrix} x_1 \\ \cdots \\ x_m \end{bmatrix} \mapsto \begin{bmatrix} x_1 \\ \cdots \\ x_{m-1} \end{bmatrix}$$

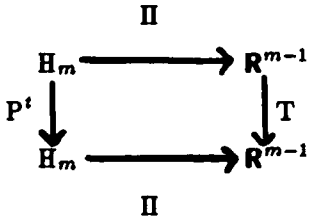
Given any m×d matrix C there is a unique affine transformation $T\text{: } \mathbf{R}^{m-1} \to \mathbf{R}^d$ making the following diagram commute.



Denote T = T(C). It can be written out in coordinate form

$$Tx = [v_1 - v_m \mid \cdots \mid v_{m-1} - v_m]x + v_m$$

where $v_1^t, \ldots, v_m^t$ are the row vectors of C. Similarly if P is an m×m row stochastic matrix then there is a unique affine transformation $T\text{: } \mathbf{R}^{m-1} \to \mathbf{R}^{m-1}$ making the following diagram commute.



In fact $T = T(P\Pi^t)$, so that the matrix part of T is given by $A = (p_{ji} - p_{mi})_{i,j=1}^{m-1}$ and the translational part of T is the column vector $(p_{mi})_{i=1}^{m-1}$.

The PI's random algorithm for generating the curve $\mathcal{A}$ follows. The inputs are the row stochastic matrices P(i) and the control polygon C. Define T(i) from P(i) via the commutative diagram above.

Algorithm (Sub-Division Methods):
initialize $\mathbf{x}$ = (the fixed point of T(0)) $\in \mathbf{R}^{m-1}$
for n = 1, 10000
    plot $T(C)\mathbf{x} \in \mathbf{R}^d$
    choose $i \in \{0, ..., N-1\}$ randomly
    $\mathbf{x} \leftarrow T(i)\mathbf{x}$

This algorithm is illustrated in Figure 4. The curves in Figs. 4a)-e) are all quadratic B-splines. They correspond to different placements of the control points v_i. Observe how this algorithm parallelizes. Instead of running an orbit of length 10000 on one processor, one can run (say) 10 processors, each generating only 1000 points of an orbit.

Wavelets are compactly supported functions W(x), $x \in \mathbf{R}$, having the special feature that the family $\{W(2^j x-k): j \geq 0, k \in \mathbf{Z}\}$ of translates and dilations are orthogonal (in the L_2-sense). These functions $W(2^j x-k)$ are local in both space and frequency, and thus afford a new type of basis for orthogonal expansions.

Daubechies' wavelets are constructed through solutions g(x), $x \in \mathbf{R}$, of certain dilation equations

$$\begin{cases} g(x) = \sum_k a_k g(2x-k) \\ \sum_k g(x-k) = 1 \end{cases} \tag{4}$$

Precisely, W is given in terms of g as

$$W(x) = \sum_k (-1)^k a_{1-k} g(2x-k) \tag{5}$$

If the coefficients a_k are chosen in a very special way, then W will have the desired orthogonality property. Precisely $a_k = 0$ for $k < 0$ or $k > m$, where $m = 2p-1$; and the 2p coefficients $a_0, \ldots, a_m$ have to satisfy the 2p conditions

$$\begin{aligned} &\Sigma a_{2k} = 1, \quad \Sigma a_{2k+1} = 1 \\ &\Sigma (-1)^k k^\ell a_k = 0; \qquad \ell = 1, \ldots, p-1 \\ &\Sigma a_k a_{k+2\ell} = 0; \qquad \ell = 1, \ldots, p-1 \end{aligned}$$

Then $W = W_p$ given by (5) will have the desired orthogonality property.

Wavelet generation can be put into the preceding framework by taking $N = 2$, and setting

$$P(0) = (a_{2j-i-1})_{i,j=1}^m, \quad P(1) = (a_{2j-i})_{i,j=1}^m; \tag{6}$$

and taking as control points

$$v_j = e_j \in \mathbf{R}^{m-1}, \quad 1 \le j \le m-1,$$

where e_j are the standard unit vectors, and

$$v_m = 0.$$

Define T(0), T(1) in terms of P(0), P(1) via the commutative diagram above. Then the PI's random algorithm for generating the solution g of the dilation equations is as follows.

Algorithm (Solution of Dilation Equations):

initialize $a = 0$, $x =$ (the fixed point of $T(0)$) $\in \mathbf{R}^{m-1}$

for $n = 1, 10000$

$x_m = 1 - \sum_{k=1}^{m-1} \Sigma x_k$

plot $(a, x_1), \ldots, (a+m-1, x_m)$

choose $i \in \{0,1\}$ randomly

$a \leftarrow \frac{1}{2}(a+i)$

$x \leftarrow T(i)x$

This is illustrated in Figures 5 and 6. The corresponding algorithm for generating wavelets, based on (5), is as follows.

Algorithm (Daubechies' Wavelets):

initialize $a = 0$, $x =$ (the fixed point of $T(0)$) $\in \mathbf{R}^{m-1}$

for $n = 1, 10000$

$$x_m = 1 - \sum_{k=1}^{m-1} x_k$$

$$\text{plot}\left(a + \frac{\ell}{2}, \sum_{k=\max(0,1-\ell)}^{\min(m,m-\ell)} (-1)^{k-1} a_k x_{k+\ell}\right), \quad -m+1 \le \ell \le m$$

choose $i \in \{0,1\}$ randomly

$a \leftarrow \frac{1}{2}(a + \frac{i}{2})$

$x \leftarrow T(i)x$

This is illustrated in Figures 7 and 8.

In all of these algorithms it is not necessary to initialize x as the fixed point of T(0). Rather one can initialize x arbitrarily (say $x = 0 \in \mathbf{R}^{m-1}$), and skip the plotting until $n = 10$ or 20. The point is that no matter how x is initialized, the IFS orbit will quickly move into the attractor. This work was published in [6], [7], [8].

What Next?

There are many important follow-up problems to attack and algorithms to develop for IFS theory. Listed below are some objectives the PI has already begun working on.

1. **Conditions ensuring ergodicity:** When generating curves using sub-division methods one is given m×m row stochastic matrices P(0), ..., P(N−1) as described above. If these matrices have all non-negative entries then under an irreducibility-type condition it can be shown that the random algorithm above for sub-division methods converges. More precisely, it can be shown that the underlying Markov chain is ergodic. In fact it suffices to show that

$$\overline{\lim_n} \frac{1}{n} \log||A(i_1) \cdots A(i_n)|| < 0 \tag{7}$$

for any sequence $(i_1,i_2, ...)$, where A(i) is the $(m-1)\times(m-1)$ matrix part of the affine map T(i) constructed above (via the commutative diagram).

On the other hand if the matrices P(i) are allowed to have some negative entries, then conditions ensuring (7) are not as apparent. In particular for line averaging schemes where P(0),P(1) are given by (6), what conditions on the coefficients a_k ensure (7)? Or if the coefficients a_k depend on some parameters θ, what are the critical values of θ for which the underlying Markov chain switches from recurrence to transience? This is the most significant and pressing question. Partial answers are given in [6, Thm. II], where conditions are presented under which there will exist some operator norm making A(0), ..., A(N−1) strict contractions. But more work needs to be done on this question.

2. **Surface generation:** The work on curve generation for sub-division methods and wavelets needs to be extended to surfaces. Surface generation, like curve generation, can be set up through refinement schemes; and the PI's technique produces random algorithms which generate the same surfaces as the recursive algorithms. This needs to be carried out.

3. **Dimension analysis:** The fractal dimensions of the curves and surfaces produced by the above random algorithms need to be determined. Fractal dimension is a major characteristic of surface roughness. In special cases the dimension of an IFS attractor is known, and this analysis needs to be extended to the setting of sub-division methods described above, involving row stochastic matrices P(0), ..., P(N−1). Knowing the dimension can also lead to determination of critical parameter values, as described in 1. above.

4. **Extreme points of attractors:** Let J be the Julia set for a complex polynomial f(z), and let E be the set of extreme points of (the convex hull of) J. As mentioned in §II above, the PI has observed in some computer-graphical experiments that $f(E) \subseteq E$. This needs to be investigated, to see whether this invariance property applies to more general dynamical systems than affine IFS.

5. **Animation encoding:** As a natural follow-up to the work on fractal animation described above in §VI, the PI has been looking into encoding for animation. Here the data compression ratios are enormous, since the encoding of two "endpoint" images suffices to generate the intermediates. Furthermore in certain respects animation encoding is easier than still image encoding. This is because a dynamic sequence of images often conveys additional information about the individual still images, such as segmentation. Velocity tracking of boundaries and key features of an image can be used to decide where to position the temporal IFS linear interpolation points (i.e., where to break the

animation up into "piecewise linear" video segments), and how to identify the images as IFS mixtures. Animation is potentially the most exciting application of IFS encoding.

$$T_1(x) = \begin{pmatrix} 0.8 & 0 \\ 0 & 0.8 \end{pmatrix} x + \begin{pmatrix} 0.1 \\ 0.04 \end{pmatrix} \qquad T_2(x) = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix} x + \begin{pmatrix} 0.25 \\ 0.4 \end{pmatrix}$$

$$T_3(x) = \begin{pmatrix} 0.355 & -0.355 \\ 0.355 & 0.355 \end{pmatrix} x + \begin{pmatrix} 0.266 \\ 0.078 \end{pmatrix} \qquad T_4(x) = \begin{pmatrix} 0.355 & 0.355 \\ -0.355 & 0.355 \end{pmatrix} x + \begin{pmatrix} 0.378 \\ 0.434 \end{pmatrix}$$

$$p_1 = 0.5 \quad p_2 = 0.168 \quad p_3 = 0.166 \quad p_4 = 0.166$$

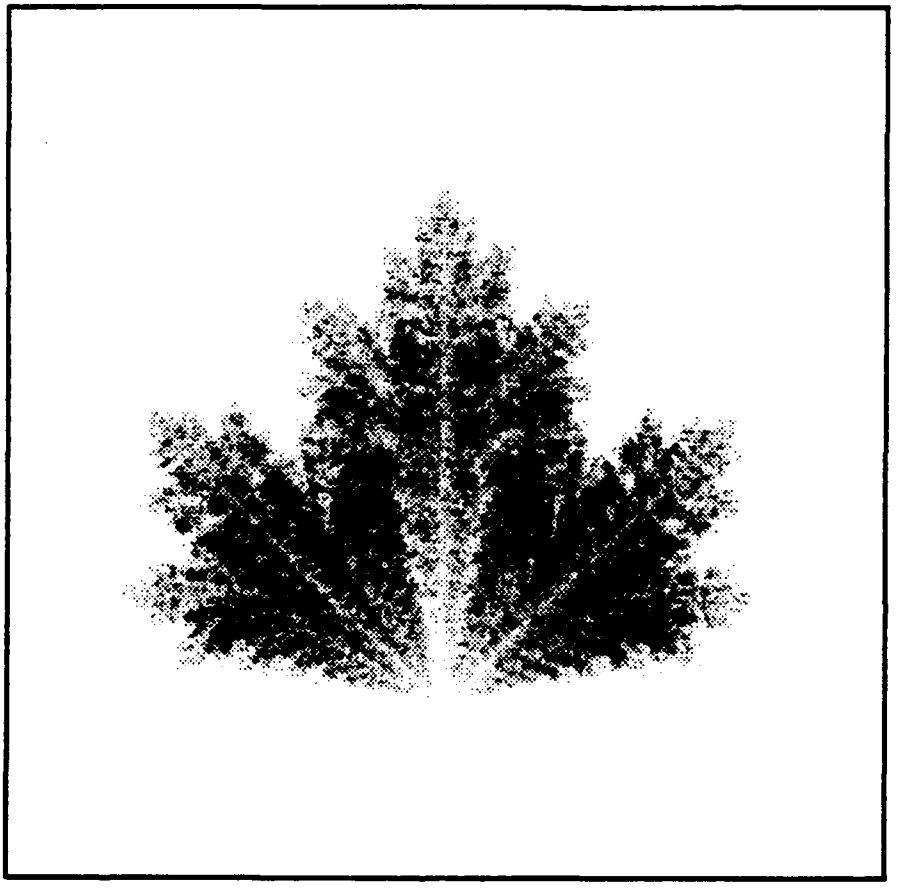


Figure 1: Maple Leaf

This image was generated using the basic IFS algorithm. The affine maps listed above were composed randomly, according to the respective probabilities shown beneath them. The darker pixels had a higher proportion of points in the orbit fall into them. (The window here is $0 \leq x,y \leq 1$.)

$$T_1(x) = \begin{pmatrix} 0.5 & 0.5 \\ -0.5 & 0.5 \end{pmatrix} x + \begin{pmatrix} 0.125 \\ 0.625 \end{pmatrix} \quad T_2(x) = \begin{pmatrix} 0.5 & 0.5 \\ -0.5 & 0.5 \end{pmatrix} x + \begin{pmatrix} -0.125 \\ 0.375 \end{pmatrix}$$

$$p_1 = 0.5 \quad p_2 = 0.5$$

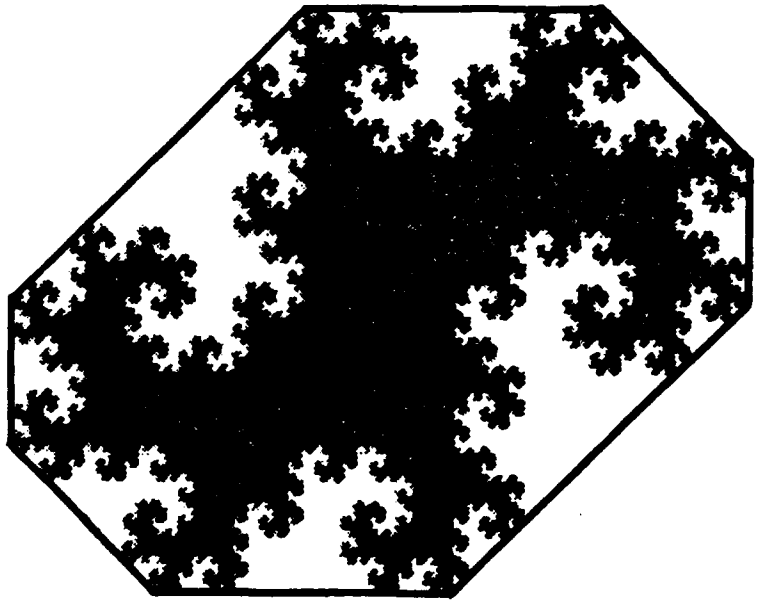


Figure 2: Convex Hull of an Attractor

Each extreme point is the image of some extreme point under T_1 or T_2. The interior angles of this convex hull are all 135°. This is because the maps T_1 and T_2 are conformal. (The window here is $-1 \leq x,y \leq 1$.)

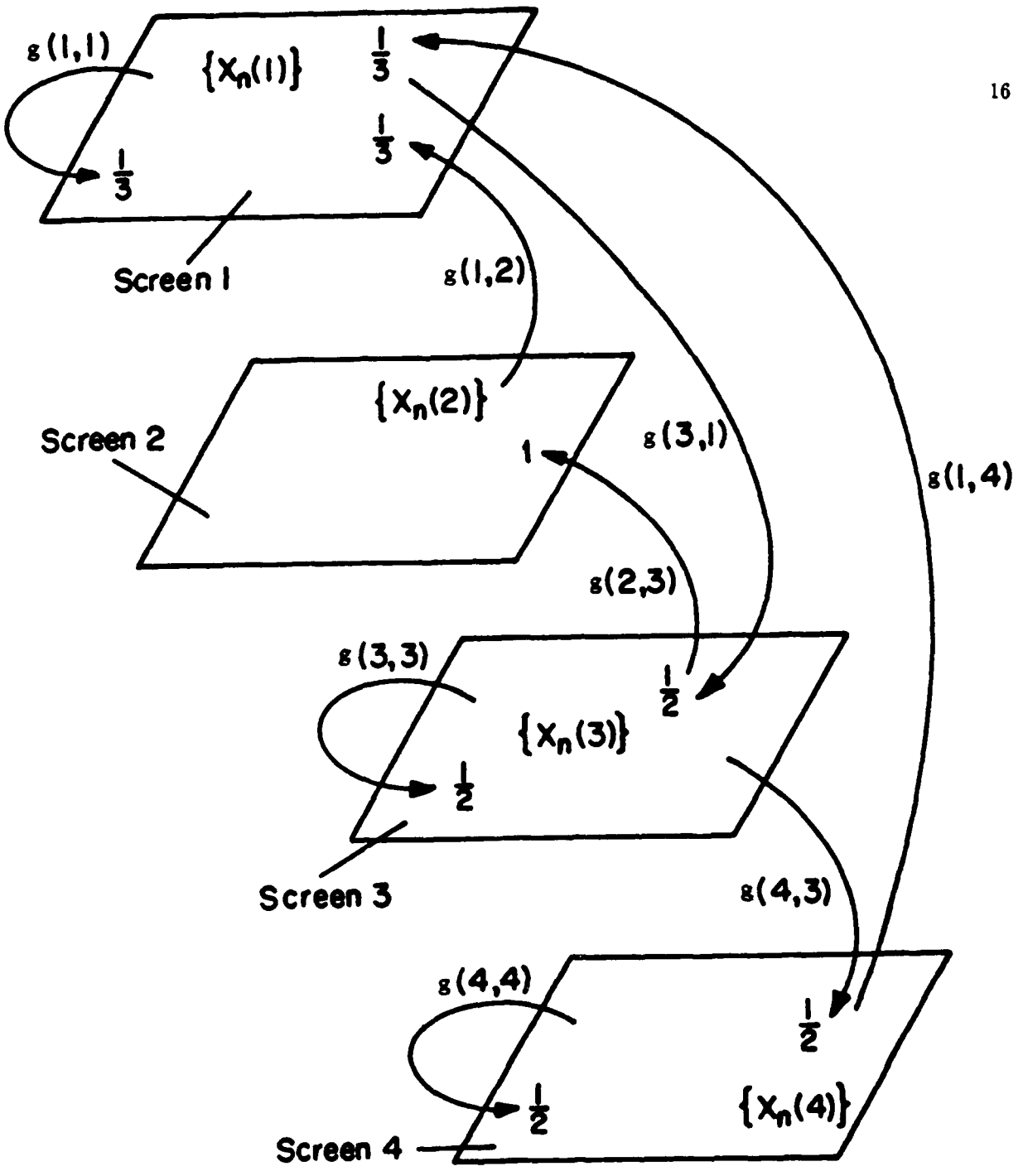


Figure 3: 4-Screen Mixing Schematic

The switching among screens is generated by the transition probability matrix

$$\begin{bmatrix} 1/3 & 1/3 & 0 & 1/3 \\ 0 & 0 & 1 & 0 \\ 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 1/2 & 1/2 \end{bmatrix}$$

read v_1, v_2, v_3, v_4

initialize $x = y = z = 0$

for $n = 1, 10000$

plot $[v_1 - v_4 \mid v_2 - v_4 \mid v_3 - v_4]x + v_4$

choose $i \in \{0,1\}$ randomly

if $i = 0$

then $x \leftarrow x + \frac{1}{2}y$

$y \leftarrow \frac{1}{2} - \frac{1}{2}x + \frac{1}{4}y + \frac{1}{4}z$

$z \leftarrow 1 - x - y$

else $z \leftarrow \frac{1}{2}x + \frac{3}{4}y + \frac{1}{2}z$

$y \leftarrow \frac{1}{2}x + \frac{1}{4}y$

$x \leftarrow 0$

Figure 4: Random Algorithm to Generate Quadratic B-Splines

Pick any points v_1, v_2, v_3, v_4 and this random algorithm produces their quadratic spline fit. This curve passes through v_1, $\frac{1}{2}v_2 + \frac{1}{2}v_3$, v_4 and is tangent to the lines v_1v_2, v_2v_3, v_3v_4 respectively, at these points. The row stochastic matrices here are given by

$$P(0) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 0 & 3/4 & 1/4 & 0 \\ 0 & 1/2 & 1/2 & 0 \end{bmatrix}, \quad P(1) = \begin{bmatrix} 0 & 1/2 & 1/2 & 0 \\ 0 & 1/4 & 3/4 & 0 \\ 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

This algorithm can be optimized by running it in parallel, breaking up the loop above into many smaller loops.

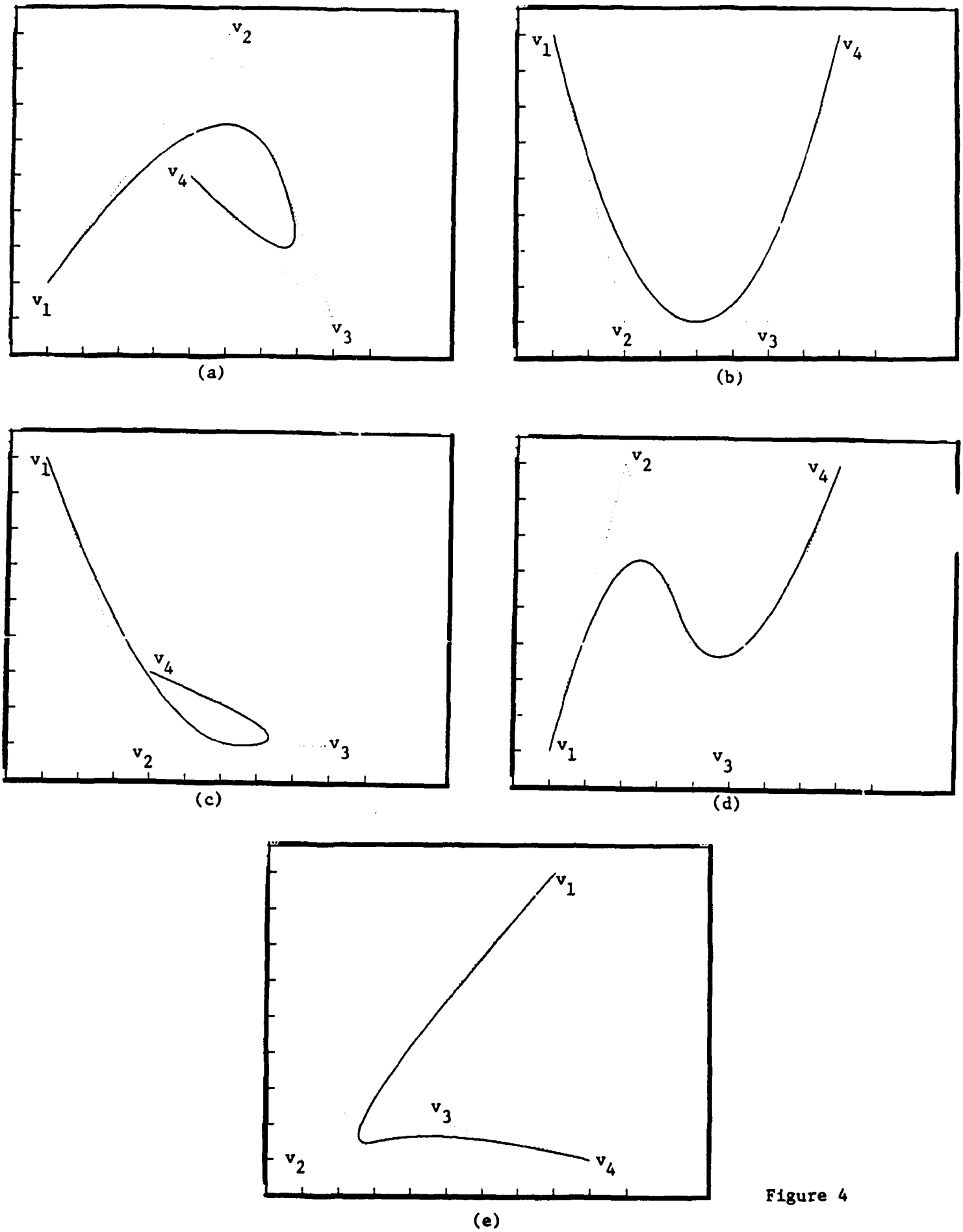


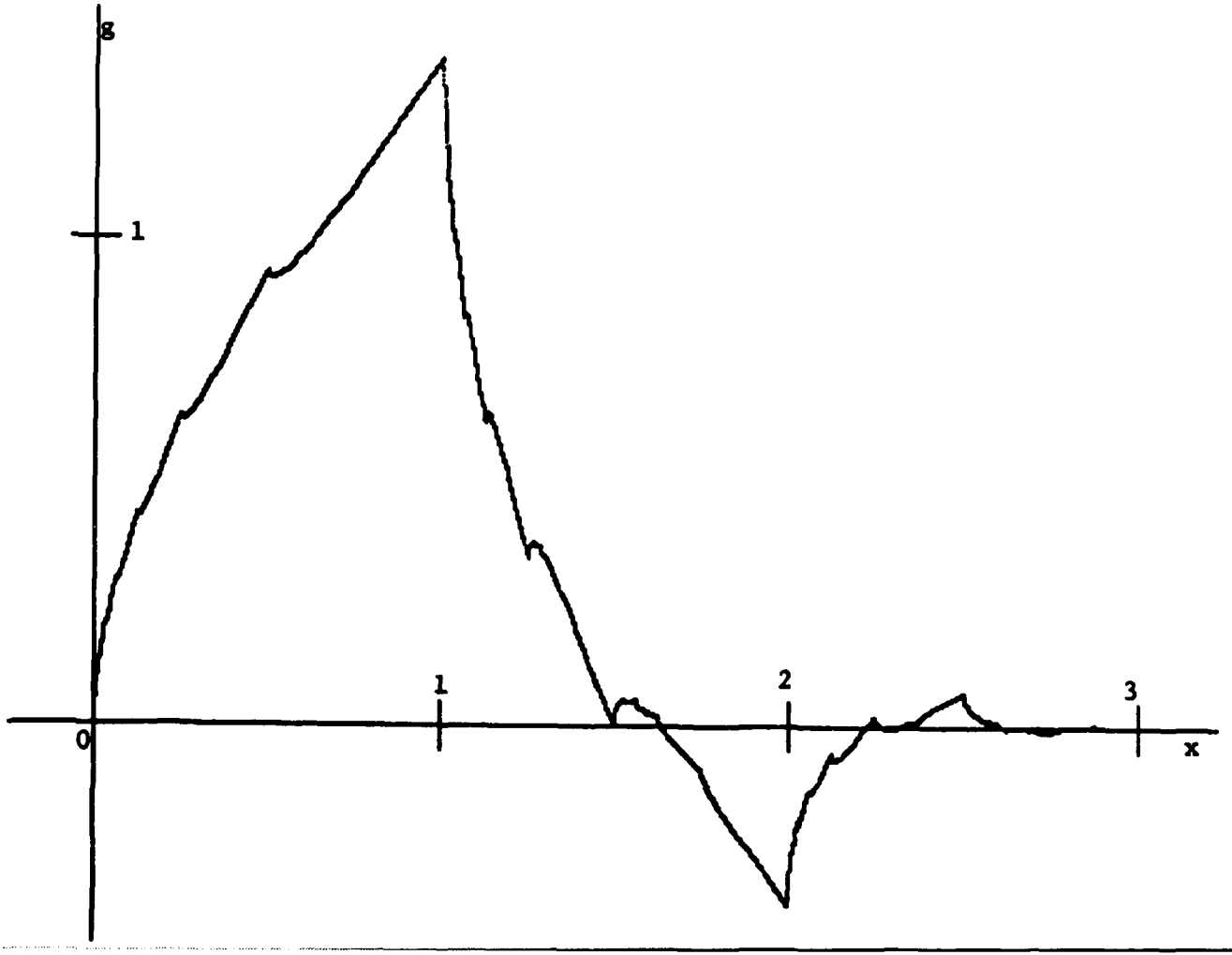
Figure 4

```
initialize a = x = 0;  y = 0.6154
for  n = 1, 10000
    z = 1 − x − y
    plot (a,x), (a+1, y), (a+2, z)
    choose i ∈ {0,1} randomly
    if i = 0
    then a ← 0.5a
         y ← 0.8 − 0.6x − 0.3y
         x ← 0.8x
    else  a ← 0.5a + 0.5
         x ← 0.5x + 0.8y
         y ← 0.5 − 0.3y
```

Figure 5: Random Algorithm to Generate the Solution of the Dilation Equations

This random algorithm plots the solution of the dilation equations (4) with (non-zero) coefficients

$$a_0 = 0.8, \quad a_1 = 0.5, \quad a_2 = 0.2, \quad a_3 = 0.5$$

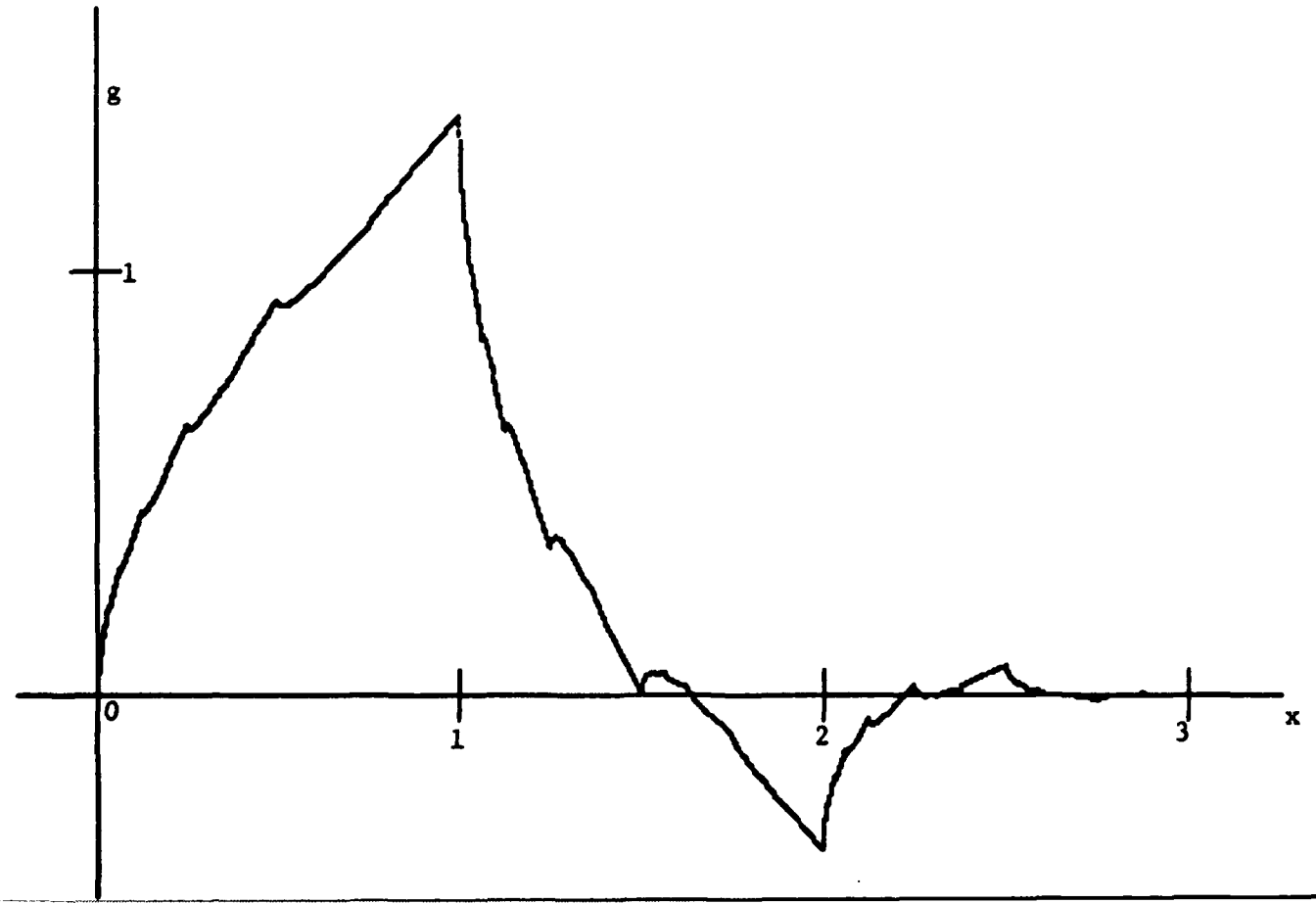


```
initialize a = x = 0;  y = 1.366
for  n = 1, 10000
    z = 1 − x − y
    plot (a,x), (a+1, y), (a+2, z)
    choose i ∈ {0,1} randomly
    if i = 0
    then a ← 0.5a
         y ← −0.366x + 0.5y + 0.683
         x ← 0.683x
    else  a ← 0.5a + 0.5
         z ← x
         x ← 1.183x + 0.683y
         y ← −1.366z − 0.866y + 1.183
```

Figure 6: Random Algorithm to Generate the Solution of the Dilation Equations

This random algorithm plots the solution of the dilation equations (4), used to construct Daubechies' wavelet W_4. The (non-zero) coefficients are

$$a_0 = \frac{1+\sqrt{3}}{4}, \quad a_1 = \frac{3+\sqrt{3}}{4}, \quad a_2 = \frac{3-\sqrt{3}}{4}, \quad a_3 = \frac{1-\sqrt{3}}{4}$$



initialize $a = x = 0$; $y = 1.366$

for $n = 1, 10000$

$b_1 = -0.183x$; $b_2 = -0.5x - 0.183y$; $b_3 = 0.866x - 0.317y - 0.183$

$b_4 = 0.5x + 1.183y - 0.5$; $b_5 = 0.683(1-x-y)$

plot $(a-1, b_1)$, $(a-0.5, b_2-b_1)$, (a, b_3-b_2), $(a+0.5, b_4-b_3)$, $(a+1, b_5-b_4)$, $(a+1.5, -b_5)$

choose $i \in \{0,1\}$ randomly

if $i = 0$

then $a \leftarrow 0.5a$

$y \leftarrow -0.366x + 0.5y + 0.683$

$x \leftarrow 0.683x$

else $a \leftarrow 0.5a + 0.25$

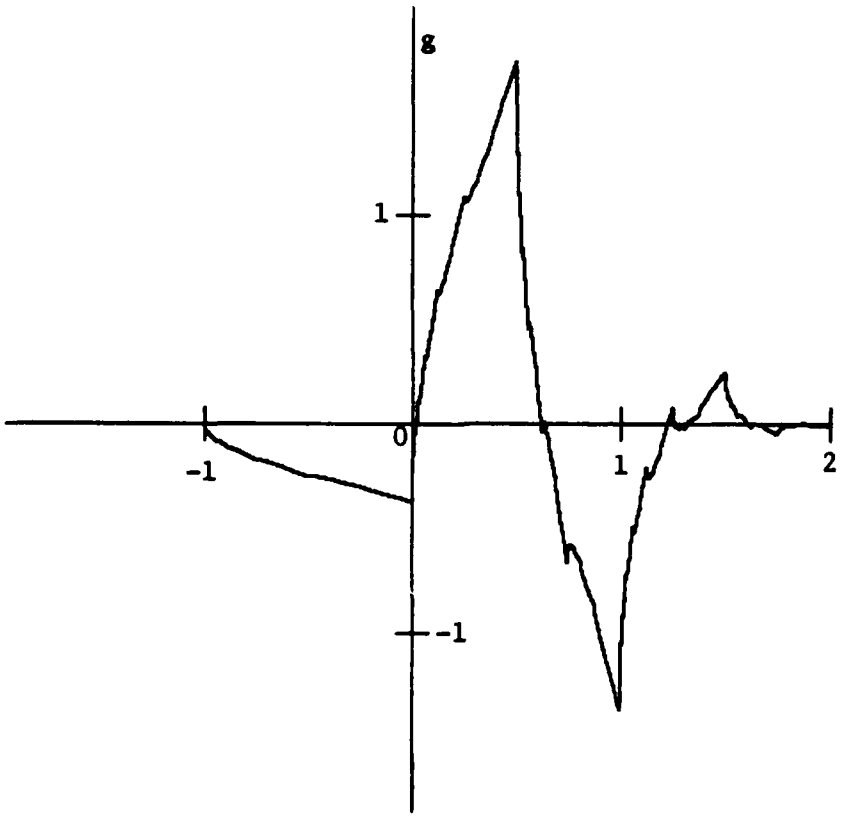
$z \leftarrow x$

$x \leftarrow 1.183x + 0.683y$

$y \leftarrow -1.366z - 0.866y + 1.183$

Figure 7: Random Algorithm to Construct Daubechies' Wavelet W_4

This wavelet is constructed out of the curve from Fig. 6 above, via (5). There is no need for a recursive tree algorithm – ergodic theory does all the running around.



initialize $r = z_0 = z_6 = x_1 = 0;\ x_2 = 1.286;\ x_3 = -0.386;\ x_4 = 0.095$

$a_0 = 0.470;\ a_1 = 1.141;\ a_2 = 0.650;\ a_3 = -0.191;\ a_4 = -0.121;\ a_5 = 0.050$

for $n = 1, 10000$

$x_5 = 1 - x_1 - x_2 - x_3 - x_4;\ a_0 \leftarrow -a_0;\ a_2 \leftarrow -a_2;\ a_4 \leftarrow -a_4$

for $\ell = -4, 5$

$m_1 = \max(0, 1-\ell),\ m_2 = \min(5, 5-\ell);\ y = 0$

For $k = m_1, m_2$

$y \leftarrow y + a_k x_{k+\ell}$

endfor

plot $(r + \frac{\ell}{2}, y)$

endfor

choose $i \in \{0,1\}$ randomly

$r \leftarrow 0.5r + 0.25i;\ a_0 \leftarrow -a_0;\ a_2 \leftarrow -a_2;\ a_4 \leftarrow -a_4$

for $\ell = 1, 5$

$z_\ell \leftarrow x_\ell;\ x_\ell \leftarrow 0$

endfor

for $\ell = 1, 4$

$m_1 = \max(0, 2\ell-6);\ m_2 = \min(5, 2\ell-1);\ j = 2\ell + i - 1$

for $k = m_1, m_2$

$x_\ell \leftarrow x_\ell + a_k z_{j-k}$

endfor

endfor

endfor

Figure 8: Random Algorithm to Construct Daubechies' Wavelet W_6

The (non-zero) coefficients a_k are given by

$a_0 = 0.470,\ a_1 = 1.141,\ a_2 = 0.650,\ a_3 = -0.191,\ a_4 = -0.121,\ a_5 = 0.050$

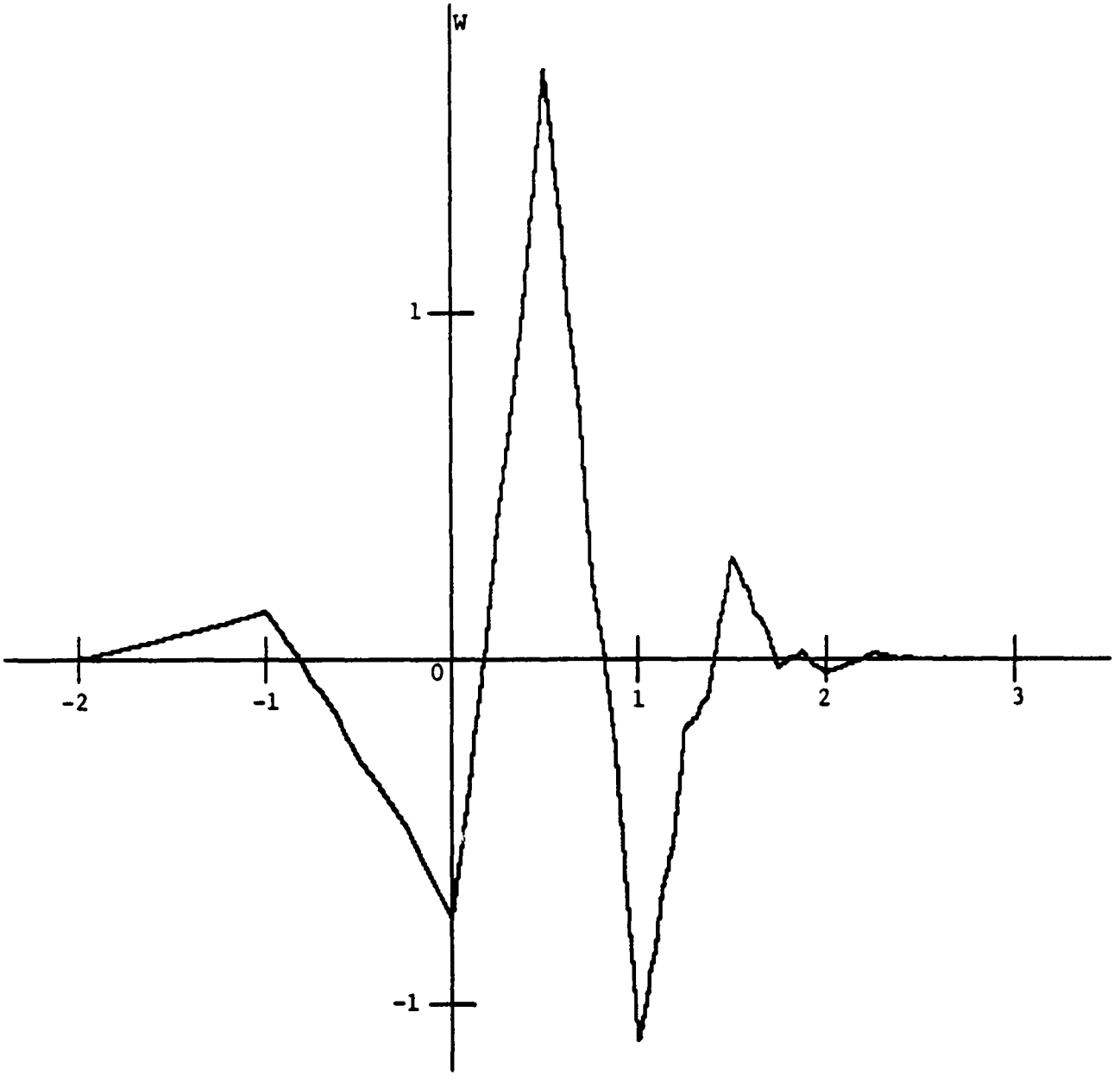


Figure 8

Published Articles under this Grant

1. Barnsley, M. F., Berger, M. A. and Soner, H. M., Mixing Markov chains and their images, *Prob. Eng. Inf. Sci.* 2 (1988), 387-414.

2. Barnsley, M. F. and Berger, M. A., Pictures worth a million words, in *Projects in Scientific Computing*: 1988-1989, Pittsburgh Supercomputing Center, 34-35.

3. Berger, M. A., Encoding images through transition probabilities, *Mathl. Comput. Modelling* 11 (1988), 575-577.

4. Berger, M. A., Images generated by orbits of 2-D Markov chains, *CHANCE* 2 (1989), 18-28.

5. Berger, M. A., Random affine IFS: mixing and encoding, submitted (Trans. Amer. Math. Soc.).

6. Berger, M. A., Random affine IFS: smooth curve generation, submitted (SIAM Review).

7. Berger, M. A., Ergodic theory, tree traversal and computer image generation, submitted (Proc. Nat'l Acad. Sci.)

8. Berger, M. A., Wavelets as attractors of dynamical systems, submitted (Bull. Amer. Math. Soc.)

9. Berger, M. A., Review of Fractals Everywhere, *Stochastics and Stochastic Reports*, in press.

10. Berger, M. A. and Soner, H. M., Random walks generated by affine mappings, *J. Theor. Prob.* 1 (1988), 239-254.

Article [2] above was prepared by science editor Michael Schneider (Pittsburgh Supercomputing Center).

Interactions under this Grant

The PI gave the following presentations:

1. Sixth International Conf. on Math. Modelling, held at Washington Univ., Aug. 10-14, 1987 (host: Dr. I. Rodin).

2. Invited talk for the seminar run by the Pittsburgh Supercomputing Center in October, 1987 (host: Dr. R. Roskies).

3. AFOSR at Bolling Air Force Base on Feb. 24, 1988 (host: Dr. A. Nachman).

4. Invited talk in Michael Barnsley's minisymposium on chaos at the annual SIAM meeting in Minneapolis, July 10-15, 1988 (host: Dr. M. Barnsley).

5. Invited talk for the seminar run by the image processing group (Grenander, McClure, Geman and Gidas) in the Division of Applied Mathematics at Brown University, Feb. 15, 1989 (host: Dr. B. Gidas).

6. NIST in Gathersburg, MD on March 29, 1989 (host: Dr. F. Sullivan).

7. NSF in Washington, DC on March 30, 1989 (host: Dr. R. Chin).

8. Special 3-day Lecture Series at Allegheny College, April 10-12, 1989.

9. Invited Address, SIAM Conference on Dynamical Systems, May 7-10, 1990 (host: Dr. H. Stech).

10. Invited talk at the meeting on Lyapunov Exponents, Mathematisches Forschungsinstitut Oberwolfach (Black Forest, Federal Republic of Germany), May 27-June 2, 1990 (host: Dr. L. Arnold, Bremen).

11. Invited talk at the First IFIP Conference on Fractals, FRACTAL 90, Lisbon, Portugal, June 6-8, 1990 (host: Dr. H.-O. Peitgen, Bremen).

Participating Professionals

Some of the PI's research on image encoding was carried out with

(1) Michael Barnsley (Prof. of Math., GA Tech.);

(2) Bill Eddy (Prof. of Statistics, CMU);

(3) John Elton (Prof. of Math., GA Tech.);

(4) Jeff Geronimo (Prof. of Math., GA Tech.);

(5) Mario Perrugia (Ph.D. student, Statistics, CMU – partially funded by AFOSR);

(6) Ron Shonkwiler (Prof. of Math., GA Tech.);

(7) Jean-Philippe Vidal (Ph.D. student, Comp. Sci., CMU – funded by AFOSR).

Some of the PI's research on mixing was carried out with

(1) Michael Barnsley;

(2) John Elton;

(3) H. Meté Soner (Prof. of Math., CMU).

Some of the PI's research on wavelets was carried out with

(1) Steve Demko (Prof. of Math., GA Tech.);

(2) Shmuel Friedland (Prof. of Math., Univ. Ill.-Chicago);

(3) Tim Kelley (M.S. student, GA Tech. – funded by AFOSR).

Computing support was provided by the Pittsburgh Supercomputing Center (CRAY Y-MP/832 and animation equipment), the Statistics Dept. at CMU (micro-Vax and animation/camera equipment), the Computer Graphics Lab in the School of Math. at GA Tech. (Masscomp 5600 Series and Encore multi-processing machine), and a SUN SPARC station in the PI's office at GA Tech.