DICTA•91 Digital Image Computing: Techniques and Applications
Royal Exhibition Building Conference Centre, Melbourne 4-6 December 1991
ed. A.]. Maeder and B.M. Jenkins *Australian Pattern Recognition Society* ISBN 0 646 07338 9

IFS ENCODING OF IMAGE SEGMENTS: NEW DETERMINISTIC SCANNING AND HYBRID ALGORITHMS FOR FAST DECODING

Harvey A. Cohen
Computer Science and Computer Engineering Department
La Trobe University
Bundoora, Victoria, 3083
Australia

ABSTRACT

Deterministic algorithms for decoding IFS (Iterated Function System) sets involves determining all the IFS (dynamic) descendants of seed pixels. Timing data was obtained for previously describe algorithms, and new algorithms: the scanning algorithm; the stack algorithm; and a hybrid combination. Decode time data indicates the superiority of the hybrid algorithm

1. INTRODUCTION

This paper reviews IFS decoding, and describes in detail new deterministic algorithms for IFS decoding, reports on the rate of decoding , and presents timing data using these algorithms for decoding four representative IFS sets, together with like data for Barnsley's Random Iteration Decoding Algorithm

1.1 Background

It has been known for a considerable time (see e.g., [1]) that an instance of what later were termed deterministic fractals [4[4] could be specified as the attractor of a set of contraction mappings. Barnsley and co-workers,[5][6] pointed out about ten years ago that ANY set in the plane, to any desired accuracy, can be approximated as the attractor of a set of contraction mappings. Subsequently, Barnsley and Sloan [7] have proposed the use of IFS [Iterated Function Systems] - sets of contraction maps of which each mapping is an affine transformation - for encoding of 'high quality colour images. Barnsley has demonstrated [7][8][9][11] examples of manually encoded IFS of exceedingly high compression combined with visually satisfying output on decoding. This work showed the potential significance of IFS image encoding to image compression and storage, with applications in broadband services, as well as to image analysis [10] and synthesis. The major problem to be solved is fully-effective automated encoding. Iterated Systems Inc has recently announced a system board for somewhat coarse-grained IFS encoding of images. It is now salient to devise new algorithms for decoding IFS parameter sets, and to evaluate such algorithms for efficiency in comparison to those previously reported. [12[13]

2 . IFS ENCODING OF IMAGE SEGMENTS

The IFS code [8] for a (two-dimensional) image segment consists of the parameters A,B,C,D,E,F of s linear mapping functions W[t], t =1 . . s Such a mapping function transforms a pixel coordinate (x,y) according to

$$\boldsymbol{W}\begin{pmatrix}\boldsymbol{x}\\ \boldsymbol{y}\end{pmatrix} = \begin{pmatrix}\boldsymbol{A} & \boldsymbol{B}\\ \boldsymbol{C} & \boldsymbol{D}\end{pmatrix}\begin{pmatrix}\boldsymbol{x}\\ \boldsymbol{y}\end{pmatrix} + \begin{pmatrix}\boldsymbol{E}\\ \boldsymbol{F}\end{pmatrix} \qquad \text{with contractivity } |AD - BC|$$

The decoding of an. IFS parameter set determines a set of pixels which is the digitised approximation to the 'attractor set $\mathcal{A}$(S) the set S of mappings.

{W[t], t =1 . . s }

In order to elucidate the algorithms for computing this (approximation to the) attractor, it is useful to introduce the concept of the IFS descendants of a pixel $\boldsymbol{p}$ = (x,y). We call the points derived from $\boldsymbol{p}$ by applying the s IFS maps once, viz

$$W_r\,\boldsymbol{p} \text{ for } r=1..s$$

the IFS sons of $\boldsymbol{p}$ Likewise the IFS sons of these points are the IFS grandsons of $\boldsymbol{p}$:

$$W_{i_2} . W_{i_1} . \boldsymbol{p}$$

for any pairs of labels $i_1 i_2$, In general, the IFS descendants of $\boldsymbol{p}$ are the pixels

$$W_{i_d} . \; . \; .. \; W . \; W_{i_4} . \; W_{i_3} . \; W_{i_2} . \; W_{i_1} p$$

for any finite set of map labels, $i_1 i_2 i_3 i_4 \ldots i_d$.

The remainder of this section is necessarily mathematical, However, using the concept of IFS descendants, decoding algorithms are detailed in Section 3. The attractor $\mathcal{A}$(S) of the set S has the property that for any $\boldsymbol{p} \in \mathcal{A}$(S) all the descendants of $\boldsymbol{p}$ lie in $\mathcal{A}$(S). Mathematically $\mathcal{A}$(S) is the closure of the set of points invariant under any finite composition of maps from S. This mathematical statement means that a descendant point may be found arbitrarily close to any point in the attractor, an important issue for the validity of digital approximations.

Williams [1] showed that the attractor can also be defined as the closure of the set of fixed points of all finite possible products of maps from S:

$$.W_{i_d} . \; . \; . \; W_{i_4} W_{i_3} W_{i_2} W_{i_1}$$

An important corollary of William's result is that the fixed points of the mappings lie in the attractor: that is, each (unique) f_t such that

$$W_t \; f_t = f_t \text{ then } f_t \in \mathcal{A}(S)$$

Note that f_t is easily computed by matrix algebra.

Hutchinson [3] showed that The IFS descendants of ANY point in the attractor of an IFS set are dense in the attractor. This result provides the underlying theory for deterministic and stochastic algorithms for the digital approximation of an IFS attractor, which is simply the decoding - to a chosen scale - of the IFS set.

3 DETERMINISTIC DECODING ALGORITHMS

The basic scheme for deterministic algorithms for the decoding of an IFS set is to compute all the IFS descendants of the seed pixel(s). (Hutchinson [3]). The natural seed pixels are the fixed points of the mappings of the IFS set.

To explicate this concept consider the case of a set described by an IFS set involving three mappings:

If there is a single seed point 0 and 3 maps in the IFS set, there are 3 immediate descendants, simply called sons, and nine grandsons. However, these descendants pixels necessarily include previously marked pixels. In analogy to breadth-first and depth-first tree searching, there are two basic algorithms schemes for deterministic algorithms, as indicated in Fig 1:

Fig 1 Sequence of marking of 2 generations of descendants from seed pixel 0 with 3 maps In the IFS set. On the left, generation-by-generation marking as for the Scanning Algorithms On the right, branch-by-branch marking as for the Stack Algorithm.

It is an inherent feature of an (unpruned) IFS descendant tree that all pixels will be redetermined at deeper levels of the tree. Dubuc and Elqortobi [12] pointed out the necessity for some form of pruning scheme, so that the descendants of a pixel are determined precisely once. These two authors give a mathematical account of the use of lists of pixel coordinates to keep a record of determined pixels. It is not clear precisely what data structure was used by Dubuc and Elqortobi in implementing these lists.
In the deterministic algorithms described here, an image array holds the iterative outcome of computation, and pixels that have been determined to lie in the attractor are 'marked' in this array. In unpruned algorithms the image array can be used to provide an indication of the increase -if any - in the number of marked

pixels, and thus determine the termination of decoding. In the pruned algorithms, the array can also be used to indicate both a newly marked pixel, and one whose descendants have been determined,

3.1 Barnsley's Deterministic Algorithm

Barnsley's Deterministic Algorithm, [8] has a similarity to Conway's Game of Life, in that it involves the use of two image arrays, one the 'current iteration' of the decoded image, the other the 'next generation' image. The current iteration array is scanned to locate marked pixels, whose sons are marked in the next generation image array, At the end of the scan, the next generation array becomes the current generation, and a 'blank' next generation array is produced. The algorithm has pedagogic interest as the seed pixels need not be chosen to lie in the attractor, as the descendants of any bounded set of pixels will ultimately lie in the attractor. Barnsley's algorithm is patently grossly inefficient, and is not amenable to pruning

3.2 Scanning Algorithms

In this paper a new class of deterministic algorithms, called scanning algorithms, is introduced in which a single augmented image array holds relevant state information for each pixel, including the information, essential for pruning, as to whether descendants of a marked pixel have already been determined. These new algorithms have the feature that during a scan, the descendants of marked pixels are marked on the (same) image array, so that the array contains a mix of generations, and the actual scanning sequence, can affect the decoding rate.

3.3 Stack Algorithms

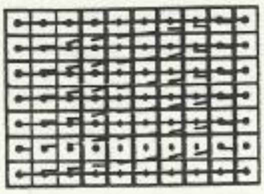
In the "stack algorithm," the descendant tree, to some specified depth, is followed in a depth first way, as indicated in Fig 1. The depth-first mode of traverse does limit stack needs compared to breadth-first traverse, but nevertheless the depth of descent is limited by stack size. In the implementations described here, the stack storage was achieved by use of recursively defined procedures. In the implementation detailed here, the image array is appropriately marked as the tree is traversed. This scheme permits a pruning scheme, whereby once a descendant is reached whose sons have been determined, that descendant limb is no longer followed.

3.4 Barnsley's Random Iteration Algorithm

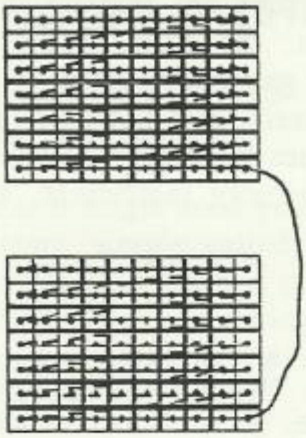
Barnsley and Demko (ref in [5]) developed what Barnsley [8] later termed the 'random iteration algorithm' in which a probability is ascribed to each mapping in an IFS set. Starting from an arbitrary pixel, only a single (Markov) chain of descendants is followed. There is no definite terminating condition implicit in the algorithm.

Figure 2. Scanning Modes: Diagrams illustrating the three modes of image scanning used in the experiments reports herein.

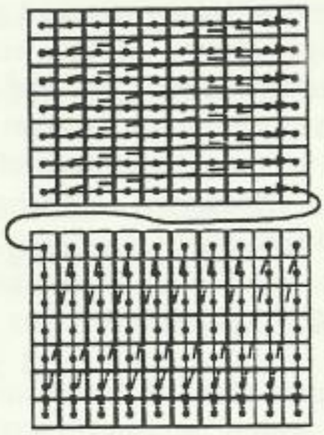
In Scanning algorithms, the entire image array (600*400 pixels herein) was scanned to locate marked pixels, and the IFS sons are marked on the array during the same scan.



TV Scan
Raster or row-col scanning



fwd-rev scanning
Involves scanning in opposite directions in consecutive Image scans



XY scanning
Row-col scan followed by col-row scan

Figure 3. - The IFS SETS STUDIED : PARAMETERS SETS

Sierpinski	A	B	C	D	E	F	P
	0.6667	0.0	0.0	.667	0.0	0.0	0.6667
	0.3333	0.00	0.00	0.333	0.3	0.4	0.1667
	0.3333	0.00	0.000	0.333	0.5	0.0	0.1666
FERN	A	B	C	D	E	F	P
	0.0001	0.0	0.0	0.16	0.0	0.0	0.01
	0.85	0.04	-.04	0.85	0.0	0.16	0.85
	0.2	-.26	0.23	0.22	0.0	0.16	0.07
	-.15	0.28	0.26	0.24	0.0	0.044	0.07
MYTREE	A	B	C	D	E	F	P
	.278	0.514	-.532	.269	0.011	0.532	0.3
	.341	-.490	0.482	.346	-.005	0.519	0.3
	.150	0.0	0.0	.520	0.0	0.0	0.2
	.140	0.0	0.0	0.190	-.005	0.019	0.2
QUAD	A	B	C	D	E	F	P
	0.500	0.0	0.0	0.50	0.0	0.0	0.25
	0.500	0.0	0.0	0.50	0.5	0.0	0.25
	0.500	0.0	0.0	0.50	0.0	0.5	0.25
	0.500	0.0	0.0	0.50	0.5	0.5	0.25

3.5 Hybrid algorithms

It takes as long to scan an image array if there is one marked pixel therein as if there are many. Hence scanning algorithms are very slow initially. In contrast, for iterative algorithms such as Barnsley's Random Iteration and the Stack Algorithm, initially there is very high efficiency, as only 'new' pixels are marked.

However Barnsley's Random Iteration Algorithm become highly inefficient when most - but not all - pixels in the attractor have been marked - as most pixels encountered on the image traverse have been previously marked - no pruning is possible. With regard to the Stack Algorithm introduced here, there are problems of stack overflow if one attempts to pursue this algorithm to such depth as would mark all pixels in an attractor. In contrast the Scanning Algorithm is effective in finding the last few unmarked pixels of an almost totally decoded IFS set. Hence, it is of interest to investigate a hybrid algorithms, with an initial Stack Algorithm Stage, followed by Scanning In the work reported here the Hybrid Algorithms involve a 10 level Stack Algorithm first stage.

QUAD FERN MYTREE SIERPINSKI

Figure 4. Decoded output of the four IFS sets used in this study

4. EXPERIMENTAL RESULTS

Timing data has been found for decoding of four IFS sets which cover a range of extremes: Quad, Fern, Sierpinski, and Mytree, whose decoded output is shown in Fig **4,** with transform data presented in Fig 3. Experiments featured:

- Implementation of code in Turbo Pascal 5.5
- Use of floating point arithmetic in computation
- Execution on a 16Mhz 3086 clone NOT equipped with a maths coprocessor
- Use of video Ram accessed by DOS routines as the image array that is scanned and marked
- Image scan areas 100* 100 [no of pixels in decoded images is tabled

Comparison of Barnsley and Six Scanning Algorithms
with scan algorithms commencing from the set of fixed points only

SOLID 1F S set used	Barnsley Random Iteration secs	TV Scan no pruning secs	TV Scan pruned secs	XY Scan no pruning secs	XY Scan with pruning secs	fwd-rev no pruning secs	fwd-rev with pruning secs
quad	44.65	21.4	*11.65*	25.65	13.89	26.65	14.65
fern	170.65	31.20	14.23	24.65	*13.18*	23.65	16,65
mytree	447.81	41.80	*18.62*	49.76	23.65	50.25	22.65
cantor	104.65	22.65	16.65	23.39	19.65	15.65	*13.65*

time denotes the fastest synthesis time for each IFS set

Comparison of Barnsley Random Iteration with Six Hybrid Algorithms
Pruned scan algorithm preceded by a 5-level pruned stacking algorithm,;
Unpruned scan algorithm preceded by a 5-level unpruned stacking algorithm.

SOLID IFS set used	Barnsley Random Iteration secs	TV Scan no pruning secs	TV Scan pruned secs	XY Scan no pruning secs	XY Scan with pruning secs	fwd-rev no pruning secs	fwd-rev with pruning secs
quad	44.65	15.65	7.65	14.34	'8.24*	15.65	9.65
fern	370.65	35.80	14.28	21.65	'10.38*	27.65	16.65
mytree	447.81	46.65	'16,13*	41.65	'16.13*	49.77	19.65
cantor	104.65	18.65	11.75	18.65	13.34	'11.10'	10.65

*time' denotes the fastest synthesis time for each IFS set

Number of pixels marked during these algorithms

SOLID IFS set used	Barnsley Random Iteration	TV Scan no pruning	TV Scan pruned	XY Scan no pruning	XY Scan with pruning	fwd-rev no pruning	fwd-rev with pruning
quad	2500	2601	2601	2601	2601	2601	2601
fern	1921	1938	1938	1938	1938	1938	1938
mytree	2024	2133	2133	2133	21.33	2133	2133
cantor	611	619	619	619	619	619	619

FIG 5 Timing Data for 100*100 Image scan region.

5. COMMENTS AND CONCLUSIONS

Image processing has been an application driven discipline, with the needs of early lunar and space exploration directing the basic formalism. The mathematics used in the established algorithms is of nineteenth century vintage. More recently, the special role of Tikhonov's regularization in explicating 'From X to Y' computer vision algorithms has been realised by this writer and others. Several workers, notably Pentland, have proposed that fractals may have a useful role in the analysis of 'natural scene' images. However, the Barnsely proposal to use IFS encoding for the compression and possibly the transmission of two dimensional images involves the first major application of modern dynamics and chaos theory to image processing. [6]

This paper, has been devoted to detailing new algorithms for decoding an IFS set, and discussing efficiency for implementation by a serial computer. The IFS sets chosen for experimental study (See fig 3,4) cover extremes of sparseness, from QUAD to SIERPINSKI so that they can be considered reasonably representative. The data collected for 100*100 image regions complements that previously presented by the author [13] for 400*600 pixel images. The data as tabled in Figure 5, together with that in [13] indicates that, at least for the IFS sets examined, a pruned scanning algorithm, with either a TV scan, or an XY-scan, is the fastest of the 'simple' algorithms. The hybrid algorithms studied, with depth 10 Stack Algorithm preceding a Scanning Algorithm, were markedly faster.

A previous paper by this writer [10] discussed the difficulties in the analysis of IFS encoded images due to non-uniqueness of the encoding. Apart from [13] the only other published account of the efficiency of IFS decoding algorithms, is that of Dubuc and Elqortobi [12], which gives timing data for various algorithms but does not give detailed implementation details. The results presented here agree with those of [12] on. the importance of pruning for speed-up, and on the slowness of Barnsley's Random Iteration Algorithm; However, in addition to our data for new algorithms, this paper highlights the significance of scanning mode for decoding, with markedly different times for different image scan modes.

6. References

1. R.F. Williams, *Compositions of contractions*, Bol Soc Brasil Mat Vol 2 (1971) pp 55-59.
2. B. B. Mandelbrot, Fractals. Form, Chance and Dimension Freeman, San Francisco (1977)
3. J.E. Hutchinson, *Fractals and Self Similarity* Indiana University Mathematics Journal, Vol 30, No 5, (1981) pp 713 -747.
4. B. B. Mandelbrot The Fractal Geometry of Nature W.H. Freeman and Co, 1982.
5. M.F. Barnsley and S.G. Demko *Iterated Function Systems and the Global Construction of Fractals* Proc. Roy. Soc. (London) A Vol A399 pp 243-275, 1983.
6. M. F. Barnsley *Making Chaotic Dynamical Systems to Order* In M. F. Barnsley and S.G. Demko (Editors) Chaotic Dynamics and Fractals , Academic Press, 1986, pp. 53-68.
7. Barnsley, M.F. and A.D. Sloan, A *Better Way to Compress Images* January 1988 *Byte* , pp. 215-223.
8. Barnsley, M. Fractals Everywhere Academic Press, 1988.
9. B.A. Cipra, *Image Capture by Computer* , Science, Vol 24, 10 March, 1989 cr.
10. Cohen, H.A. *The Application of IFS (Iterated Function Systems) to Image Analysis* Proceedings, IEEE International Conference on Image Processing ICIP'89, Sept 89, Singapore, Vol 2, pp 583-587.
11. D. Barker, *System Uses Fractals to Compress Images.* in Microbytes Column of *Byte* February, 1991 p28 & p32.
12. Dubuc, S. and A. Elqortobi, *Approximations of Fractal Sets* Journal of Computational and Applied Mathematics, Vol 29, (1990) pp 79-89.
13. H. A. Cohen , *IFS [iterated Function Systems) Encoding of Image Segments Efficiency of Decoding Algorithms,* Proc. Australian Broadband Switching and Services Symposium 91 Vol 2, pp 242-250.