DETERMINISTIC SCANNING AND HYBRID ALGORITHMS FOR FAST DECODING OF IFS [ITERATED FUNCTION SYSTEM] ENCODED IMAGE SETS

Harvey A. Cohen

Computer Science and Computer Engineering Department
La Trobe University
Bundoora, Victoria, Australia 3083

ABSTRACT

Deterministic algorithms for decoding IFS (Iterated Function System) sets involves determining all the IFS (dynamic) descendants of seed pixels. Realistic algorithms require pruning of previously encountered pixels on the descendant tree. Timing data was obtained for the well-known Random Iteration Algorithm, and new deterministic algorithms: the scanning algorithm; the stack algorithm; and a hybrid combination. Decode time data indicates the superiority of the pruned hybrid algorithm.

1. INTRODUCTION

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 paper first reviews IFS encoding and decoding, emphasises the need for pruning, 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.

2 IFS ENCODING

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

$$W_t \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} A B \\ C D \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} E \\ F \end{pmatrix}$$

with contractivity | AD - BC | < 1 The decoding of an IFS parameter set determines a set of pixels which is the digitised approximation to the 'attractor set' A(S) of the set S of mappings:

$$S = \{W[t], t = 1 . . s \}$$

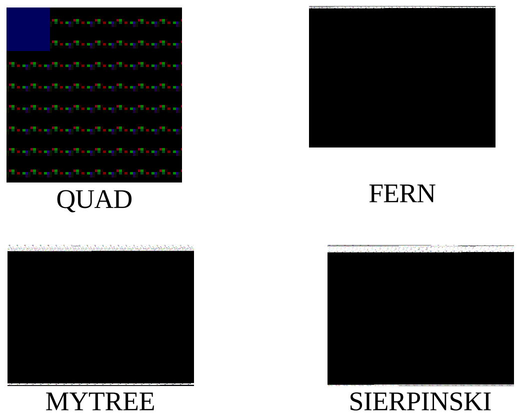


Figure 1. Decoded output of the four IFS sets in study.

Sierp	A	B	C	D	E	F	P
	0.666	0.0	0.0	.667	0.0	0.0	0.666
	0.333	0.00	0.00	0.333	0.3	0.4	0.166
	0.333	0.00	0.000	0.333	0.5	0.0	0.166
Fern	A	B	C	D	E	F	P
	0.000	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

Figure 2 - The IFS parameter sets utilised in this study.

Pre-print version of Harvey A. Cohen, *Deterministic Scanning and Hybrid Algorithms for Fast Decoding of IFS Encoded Image Sets,* Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'92, San Francisco, March 20-3, 1992, Vol III pp 509-512

2.1 IFS Descendants

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 at **x** = (x,y). We call the points derived from **x** by applying the s IFS maps once, viz

W[r]**x** for r=1..s

the IFS sons of **x**. Likewise the IFS sons of these points W[r]W[s] **x** are the IFS grandsons of **x** :for any pairs of labels r,s. In general, the IFS descendants of **x** are the pixels

W[q] . . W[s] **x**

for any finite set of map labels q . . s.

3 DETERMINISTIC DECODING

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 3 and 4.

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 'Two generation 'Deterministic Algorithm

Barnsley [8] detailed what he termed 'the deterministic algorithm' and is here termed Barnsley's 'Two generation' deterministic algorithm. This algorithm requires 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.

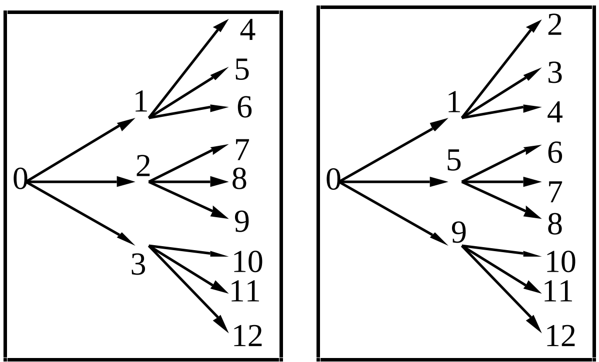


Fig 3 Sequence of marking of 2 generations of descendants from seed pixel 0 with 3 maps in the IFS set. On the left, generation-by-generation marking as for the Scanning Algorithms. On the right, branch-by-branch marking as for the Stack Algorithm.

3.3 Stack Algorithms

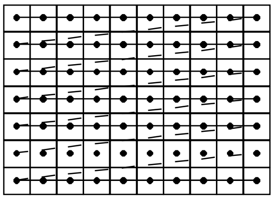
In the "stack algorithm," the descendant tree, to some specified depth, is followed in a depth first way, as indicated in Fig 1. The depth-first mode of traverse does limit stack needs compared to breadth-first traverse, but

Pre-print version of Harvey A. Cohen, *Deterministic Scanning and Hybrid Algorithms for Fast Decoding of IFS Encoded Image Sets,* Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'92, San Francisco, March 20-3, 1992, Vol III pp 509-512

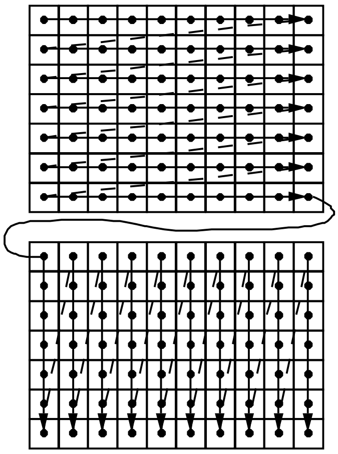
nevertheless the depth of descent is limited by stack size. In the implementations described here, the stack storage was achieved by use of recursively defined procedures. In the implementation detailed here, the image array is appropriately marked as the tree is traversed. This scheme permits a pruning scheme, whereby once a descendant is reached whose sons have been determined, that descendant limb is no longer followed.

3.4 The Role of Scanning Mode

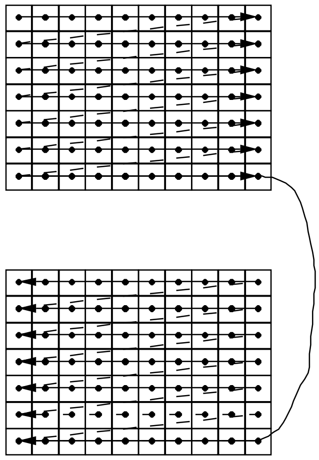
In scanning algorithms, the whole image is scanned, and IFS descendants marked (and possibly) remarked during the scan. The question then arises, does the manner of scan influence the efficiency of the algorithm. To seek answers to this question, three scanning modes were applied:



TV Scan



XY scanning



fwd-rev scanning

Figure 4. Image Scan Modes used in the experiments.

3.5 The Role of Pruning

Deterministic algorithms involve computing all the IFS descendants starting from seed pixels. In a scanning algorithm, once the IFS sons of a pixel in the attractor have been marked, there is no purpose in in determining and marking these pixels on subsequent scans. What needs to be done is to mark 'new' pixels as 'fringe' and 'done', and to only determine the IFS sons of 'fringe' pixels. Likewise the stack algorithm is pruned by reference to the image array.

3.6 The 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. In this study a count of pixels marked was made every 10000 iterations, and if no increase in pixels marked then process was stopped. This terminating condition involves the cost of the counting scan.

3.7 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.

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 1, with transform data presented in Fig 2. Data was obtained for scanning and hybrid algorithms, and also for comparison Barnsley's Random Iteration Algorithm. For scanning algorithms, the scanning modes TV, XY, and fwd-rev of Fig 4 were used. The experiments were performed on an a 16 Mhz 3086 PC, not equipped with a 3087 coprocessor. Code was implemented in Turbo Pascal 5.5. The Image Array was in video RAM and was accessed by DOS routines.

<u>Comparison of Deterministic and Random Iteration</u>

Scan algorithms commencing from fixed point set only

IFS set used	Barnsley Random Iteration secs	TV Scan not pruning secs	TV Scan pruned secs	XY Scan not pruned secs	fwd_rev Scan no_pru secs	XY Scan pruned secs
Quad	921	777	*367*	1273	895	394
Fern	3870	1212	545	1223	821	*333*
Mytree	9301	6493	*405*	1228	2964	541
Sierp...	1305	541	*481*	1057	521	500

time denotes the fastest synthesis time for each IFS set.

<u>Comparison of Hybrid and Random Iteration</u>

• 10-fold pruned non-scanningdeterministic algorithm: + pruned scan algorithms

• 10-fold unpruned non-scanning deterministic algorithm: + unpruned scan algorithms

IFS set used	Barnsley Random Iteration secs	TV Scan not pruned secs	TV Scan pruned secs	XY Scan no pruning secs	fwd_rev Scan no_prun secs	XY Scan with pruning secs
Quad	917	1904	*195*	433	2021	247
Fern	3851	3443	512	1311	6649	*331*
Mytree	9347	1097	555	1645	1082	*544*
Sierp	1299	446	331	723	379	*324*

time denotes the fastest synthesis time for each set

Number of pixels marked during these algorithms:

IFS set used	Barnsley Random Iteration	TV Scan no pruning	TV Scan pruned	XY Scan no pruning	fwd_rev Scan no_prun	XY Scan with pruning
Quad	72413	72960	72960	72960	72960	72960
Fern	44250	44785	44785	44785	44785	44785
Mytree	31689	33846	33979	33846	33846	33848
Sierp	7344	7547	7547	7547	7547	7547

FIG 5. Timing Data and number of pixels marked for 400*600 Image scan region.

5. COMMENTS AND CONCLUSIONS

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 1,2) cover extremes of sparseness, from QUAD to SIERPINSKI so that they can be considered reasonably representative. The data was collected for 400*600 image regions and the IFS parameters were scaled to give the pixel sizes indicated. The data s tabled in Figure 5. This data, together with other data [9] and data to be presented elsewhere, indicates that for unpruned algorithms, scanning mode makes major differences in speed, but that, for unpruned algorithms, either TV scan, or the more complex XY scan, are generally equally effective. Pruned algorithms are considerably faster than unpruned algorithms, and there is further sizeable speed-up for hybrid algorithms. The hybrid algorithms studied, with depth 10 Stack Algorithm preceding a Scanning Algorithm, were thus found the fastest, with scanning mode apparently not usually important.

A previous paper by this writer [7] discussed the difficulties in the analysis of IFS encoded images due to non-uniqueness of the encoding. Apart from [9] the only other published account of the efficiency of IFS decoding algorithms, is that of Dubuc and Elqortobi [8], which gives timing data for various algorithms but does not give detailed implementation details. The results presented here agree with those of [8] 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] J.E. Hutchinson, *Fractals and Self Similarity* Indiana University Mathematics Journal, Vol 30, No 5, (1981) pp 713 -747.

Pre-print version of Harvey A. Cohen, *Deterministic Scanning and Hybrid Algorithms for Fast Decoding of IFS Encoded Image Sets*, Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'92, San Francisco, March 20-3, 1992, Vol III pp 509-512

[3] B.B.Mandelbrot The Fractal Geometry of Nature W.H. Freeman and Co, 1982.

[4] 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.

[5] Barnsley, M.F. and A.D. Sloan, *A Better Way to Compress Images* , January 1988 *Byte* , pp.215-223.

[6] Barnsley, M. Fractals Everywhere Academic Press, 1988.

[7] 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.

[8] Dubuc, S. and A. Elqortobi, *Approximations of Fractal Sets* Journal of Computational and Applied Mathematics, Vol 29, (1990) pp 79-89.

[9] 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.

7. ACKNOWLEDGEMENTS

The research described here was performed while the author was a Telecom (Australia) Short-Term Research Fellow. Partial support of the Australian Telecommunications and Electronics Research Board is also acknowledged.

APPENDIX

Comparison of Deterministic and Random Iteration

Scan algorithms commencing from fixed point set only

IFS set used	Barnsley Random Iteration secs	TV Scan not pruning secs	TV Scan pruned secs	XY Scan not pruned secs	fwd_rev Scan no_pru secs	XY Scan pruned secs
Quad	921	777	*367*	1273	895	394
Fern	3870	1212	545	1223	821	*333*
Mytree	9301	6493	*405*	1228	2964	541
Sierp...	1305	541	*481*	1057	521	500

time denotes the fastest synthesis time for each IFS set.

Comparison of Hybrid and Random Iteration

• 10-fold pruned non-scanningdeterministic algorithm: + pruned scan algorithms

• 10-fold unpruned non-scanning deterministic algorithm: + unpruned scan algorithms

IFS set used	Barnsley Random Iteration secs	TV Scan not pruned secs	TV Scan pruned secs	XY Scan no pruning secs	fwd_rev Scan no_prun secs	XY Scan with pruning secs
Quad	917	1904	*195*	433	2021	247
Fern	3851	3443	512	1311	6649	*331*
Mytree	9347	1097	555	1645	1082	*544*
Sierp	1299	446	331	723	379	*324*

time denotes the fastest synthesis time for each set

Number of pixels marked during these algorithms:

IFS set used	Barnsley Random Iteration	TV Scan no pruning	TV Scan pruned	XY Scan no pruning	fwd_rev Scan no_prun	XY Scan with pruning
Quad	72413	72960	72960	72960	72960	72960
Fern	44250	44785	44785	44785	44785	44785
Mytree	31689	33846	33979	33846	33846	33848
Sierp	7344	7547	7547	7547	7547	7547

FIG 6 Timing Data for 400*600 Image scan region.