Harvey A. Cohen, *Map Colour Rendering of IFS Fractals*, Proceedings, Australian Pattern Recognition Society Workshop on Colour Imaging and Applications, Canberra 5-7 December, 1994, pp 43-48. This is the preprint.

MAP-COLOUR RENDERING OF IFS FRACTALS

Harvey A. Cohen
Computer Science and Computer Engineering, La Trobe University, Bundoora Victoria Australia 3083
Email: H.Cohen@latrobe.edu.au

ABSTRACT

In this paper a simple scheme of map colour is introduced for the colour rendering of IFS fractals. The way of incorporating it into Barnsley's random iteration and versions of Hutchinson's deterministic algorithm is detailed. Examples are presented showing an elegant stylised colouring of such fractals as the Barnsley Fern and the Sierpinski gasket that provide attractive motifs for visualization purposes.

ITERATED FUNCTION SYSTEM FRACTALS

The category of fractals we are concerned with is that of deterministic fractals mathematically specified as the 'attractor' of a set S of N contraction maps such as W_i (i =0..N-1) : with

$$S = \{W_0, W_1, W_2, W_3, W_4, W_5, \ldots W_{N-1}\}.$$

See {1][2][3][4][5]. For the case where the contraction maps are affine transformations, as per

$$W\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} \qquad \text{with} \quad |ad - bc| < 1.$$

where (x,y) is the co-ordinate pair of a pixel in the plane. The nomenclature was introduced by Barnsley et al [3][4] of referring to the set of transformations as an IFS [Iterated Function System], although in his recent book [5] Barnsley also uses the term IFS to refer to general sets of contraction maps.

In this paper we describe formally a mapping scheme called map colour, and first give a global appreciation of how it serves to map subsets of an attractor. Then we present practical schemes for applying this colour rendering in conjunction with various algorithms for generating IFS fractals.

MAP COLOUR SYSTEM - GLOBAL PERSPECTIVE

In this section a mathematical perspective of the IFS set colouring is outlined that depends on attributing a number colour, colour(r) to map W_r from the IFS Set. Formally the attractor $A(S)$, is the (closure of the) set of points invariant under any finite composition of maps from S, This implies that

$$W_r\,[A(S)\,] \supseteq A(S)$$

So that $A(S)$ may be expressed as the union

$$W_1\,[A(S)\,] \cup W_2\,[A(S)\,] \cup \ldots \cup W_N\,[A(S)\,]$$

The simplest version of the fractal colouring scheme to be introduced involves associating a colour with each mapping, so that each subset of the attractor has a distinct colour: $W_r\,[A(S)\,]$ has colour (r).

Where the $W_r\,[A(S)\,]$ are disjoint the colouring so induced is unique. In general, however, these sub-sets are not disjoint, and we specify that a given pixel is given a colour depending on a ranking of the possibly more than one colour that can be attributed to it.

The above colouring scheme is here termed one-level colour. This is just the start of the colouring scheme. With more elaborate formalism, the attractor $A(S)$ may be expressed as the union

$$\underset{r,s}{\approx} W_r\,[W_s\,[A(S)\,]]$$

To $W_r\,[W_s\,[A(S)\,]]$ we ascribe a two-level colour according to the formula

$$\text{colour}(r,s) = N^* \text{colour}(r) + \text{colour}(s)$$

Where there is an ambiguity in the colour of a particular pixel, colour ranking determines the final colour. Note that a subset of the image W_s [A] has a unique colour under one level colour - but under two-level colour it will be broken down into sub-sets each of which is coloured in accordance with the given formula. This colouring scheme is clearly extendable to an arbitrary level.

The n-level colouring scheme as described here requires that the sets of pixels constituting a digital approximation to the attractor, and then determining the various sub-sets of the attractor by a combination of mappings, then ascribing to the pixels in these sub-sets the colour as per the formula.

A trivial example of such colouring is provided by the attractor of the set of four maps:

$$W_0\begin{pmatrix}x\\y\end{pmatrix} = \begin{pmatrix}0.5 & 0\\ 0 & 0.5\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix} \qquad W_1\begin{pmatrix}x\\y\end{pmatrix} = \begin{pmatrix}0.5 & 0\\ 0 & 0.5\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix} + \begin{pmatrix}0.5\\0\end{pmatrix}$$

$$W_2\begin{pmatrix}x\\y\end{pmatrix} = \begin{pmatrix}0.5 & 0\\ 0 & 0.5\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix} + \begin{pmatrix}0\\0.5\end{pmatrix} \qquad W_3\begin{pmatrix}x\\y\end{pmatrix} = \begin{pmatrix}0.5 & 0\\ 0 & 0.5\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix} + \begin{pmatrix}0.5\\0.5\end{pmatrix}$$

The attractor A(S) of these four maps is simply the unit square (0<x<1, 0<y<1). It is easy to see that W_0 maps pixels within this square attractor W_0 (A(S)) into the quadrant, 0<x<0.5 0<y<0.5, and that to each of the four mappings there is a unique quadrant, as per the diagrams of Fig 1.

2	3
0	1

(a)

22	23	32	33
20	31	30	31
02	03	12	13
00	01	10	11

(b)

Fig 1. Global labelling of square by the mappings W_r (r=0,1,2,3) above.

(a) The quadrants W_r(A) of the square A . . (b) The 16 regions $W_s W_r$(A) that are a disjoint sub-division of the square A. This map labelling leads to the map colour described here.

One level colouring will colour each quadrant of this attractor, while two-level colouring will indicate the 16 quadrants of these quadrants. We must stress that this is a deliberately trivial example.

MAP COLOUR SYSTEM - VIA SYNTHESIS

In the previous section it was shown how map colour could be applied to an already determined IFS set. In this section the means of applying map colour to pixels during the fractal synthesis process is described.

Random Iteration Algorithm

The random iteration algorithm for the synthesis of IFS fractals was introduced by Barnsley and co-workers in two classic papers. [3][4]. This algorithm requires that associated with each contractive map W_r (r =0,1,2,...N-1) from the IFS set S is a probability p_r, the sum of these N probabilities being 1.

For the map colour version of this algorithm, a numeric colour(r) is assigned to each of the N maps, and M, the number of colours to be utilised M specified to be a power of N, i.e., $M = N.^m$ for m >0

In the nomenclature introduced above, this system is called m-level colour.

The colour algorithm applied to random iteration proceeds as follows:

Starting from **x** anywhere in the plane, chose at random in accord with the assigned probabilities, one of the maps W_r and proceed to **y** = W_r (**x**), to which is assigned a phantom mark colour, mark_colour = colour(r).

Chose at random in accord with the probabilities, one of the maps W_t and proceed to W_t (**y**).

Set mark_colour = mark_colour*N + colour(t) (mod M)

Repeat about 50 times without actually marking any pixels. Thenceforth, continue but at each pixel actually mark it with the colour

pixel colour = mark_colour

Note that a particular pixel may be 'visited' during the random iteration more than once, and possibly assigned different colours on successive visits. A consistent colour can be maintained by assigning to each of the M colours a ranking, and only ever remarking a pixel with a colour of higher rank.

It is worth mentioning that in his textbook Fractals Everywhere [5] Barnsley describes a very inefficient colour algorithm, wherein during the random iteration a count is located at each pixel, indicating how many times that pixel was visited. Hence in order to use M colours, the random iteration algorithm must proceed long enough for some pixels to be visited M times, so to reliably apply this method a multiple of the minimum number of iterations to mark all pixels of the set is required. In contrast, map colour determines pixel colour with no extra costs.

Hutchinson's Algorithm

Hutchinson's algorithm is a conceptual deterministic algorithm for generating the attractor of an IFS set. [2]. Issues related to effective utilisation of this algorithm were first raised by Dubuc and Elqortobi, [10] while this writer has described efficient and fast algorithms based (ultimately) on Hutchinson's algorithm. [6][7]. To state the algorithm simply requires the concept of IFS descendants of a point **x** as being the set of N points

$$S(\mathbf{x}) = \{W_1(\mathbf{x}) , W_2(\mathbf{x}) ,W_3(\mathbf{x}) ,W_4(\mathbf{x}) ,W_5(\mathbf{x}) ,....W_N(\mathbf{x}) ,)\}.$$

The algorithm involves starting at one or more know points in the attractor. For example the fixed points $\mathbf{p}_r$ of the maps,

$$W_r(\mathbf{p}_r) = \mathbf{p}_r$$

Hutchinsons algorithm involves iteratively determining the descendants of the initial set, until no more descendants are determined.

We refer the reader to [7] [for a discussion of means to 'prune' such algorithms, so that the descendants of a marked pixel are determined just once. It is the colouring of pixels that permits simple pruning schemes.

Map colouring applied to Hutchinson's Algorithm, works as follows:
The fixed points $\mathbf{p}_r$ are marked with colour (r). The descendants of pixel y with colour(y) - are marked - but only if not previously marked - with colour are determined one by one in order of increasing r, with the descendant $W_r(\mathbf{y})$ being marked with pixel colour

N*colour(y) + colour(r) (mod M)

To explicate this concept consider (cf [7]) 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.

Fig 1 demonstrates two extreme variants of actual implementations of Hutchinson's algorithm, termed the Scanning and the Stack Algorithm. In the implementations of these two computer algorithms described in [7], the Scanning Algorithm has the feature that the coordinates of marked pixels are located ' naturally' within an image region, so that marked pixels are detected by scanning the entire image region. On the other hand, the Stack Algorithm in [7] was implemented by simple use of a push-down stack. The Stack Algorithm ifs highly efficient for a few iterations, whereas the scanning algorithm is highly inefficient at start-up, as whole image has to be scanned when only a few pixels are present. In [7] results established the merits of a hybrid approach, with a set number of levels of the Stack Algorithm, followed by the Scanning Algorithm.

Examples

The following examples have been produced with a 16 colour EGA pallete. For those cases where 4 maps lie in the IFS set, we impose a restriction, in order to have a close approximation to map colour with M=16, we compute mark_colour by formula given, but then:

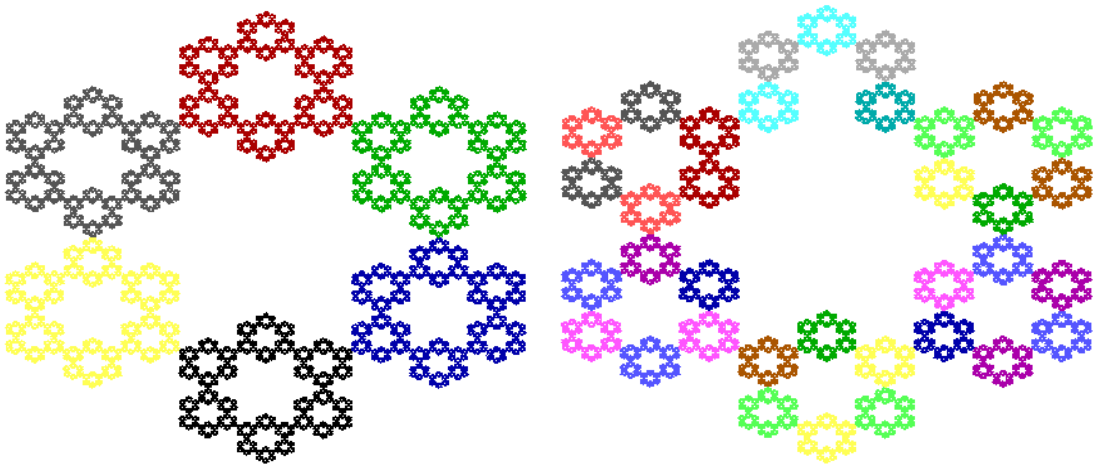
if mark_colour == WHITE then mark_colour= RED.

CONCLUSIONS

Map colour is a simple scheme for colouring, in a mathematically and aesthetically interesting way, the deterministic fractals images that are the digital approximations of the attractors of an IFS set of affine maps. The method can clearly be simply extended to other related fractals.

REFERENCES

1. Williams, R.F. *Compositions of contractions,* Bol Soc Brasil Mat Vol 2 (1971) pp 55-59.
2. Hutchinson, J.E. *Fractals and Self Similarity* Indiana University Mathematics Journal, Vol 30, No 5, (1981) pp 713 -747.
3. Barnsley, M.F. and S.G. Demko *Iterated Function Systems and the Global Construction of Fractals* Proc. Roy. Soc. A Vol A399 pp 243-275, 1983.
4. Barnsley, M.F. V. Ervin, D.Hardin and J. Lancaster *Solution of an inverse problem for fractals and other sets* Proc. Natl. Acad. Sci. U.S.A., Vol 83, 1986.
5. Barnsley, M. *Fractals Everywhere* Academic Press, 1988.
6 H. A. Cohen, *IFS [Iterated Function Systems] Encoding of Image Segments: Efficiency of Decoding Algorithms,* Proc Australian Broadband Switching and Services Symposium '91, Sydney, July 1-3, 1991, Vol 2, pp 242-250.
7 H. A. Cohen, *Deterministic Scanning and Hybrid Algorithms for Fast Decoding of IFS Encoded Image Sets,* Proc IEEE Int'l Conf on Acoustics, Speech, Signal Processing, ICASSP'92, San Francisco, March 1992, Vol III pp 509-512.
10 S. Dubuc and A. Elqortobi, *Approximations of Fractal Sets* Journal of Computational and Applied Mathematics, Vol 29, (1990) pp 79-89.



(a) (b)

Fig 2 : The figure Magen coloured (a) using 1-level colour (b) using 2-level colour. Due to use of VGA colours, colour 15 (white) has been changed to 4.

Fig 3 Barnsley Fern - which involves 4 maps, coloured with 2-level colour.[6][7]

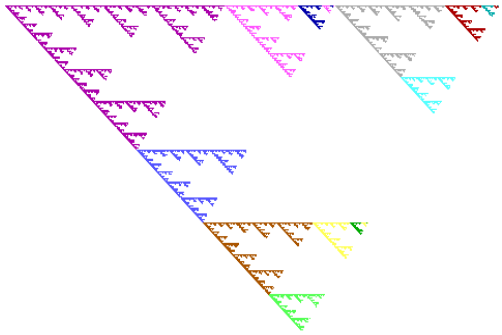
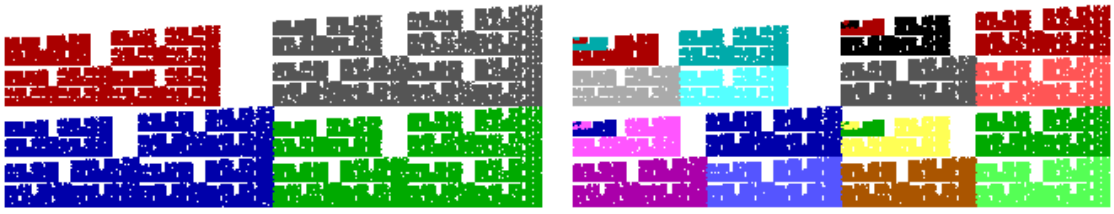


Fig 4: Sierpinski Gasket [6][7] with 2-level colour.



(a) 1-level colour (b) 2-level colour

Fig 5. Castle fractal shown with 1- and 2-level colour. Generated using random iteration algorithm prematurely stopped - yielding pleasing ragged look. Fractal code in Table 5.2.1, of Barnsley's *Fractals Everywhere* [5]