ERGODIC THEORY, FRACTAL TOPS AND COLOUR STEALING

MICHAEL BARNSLEY

Abstract. A new structure that may be associated with IFS and superIFS is described. In computer graphics applications this structure can be rendered using a new algorithm, called the "colour stealing".

1. Ergodic Theory and Fractal Tops

The goal of this lecture is to describe informally some recent realizations and work in progress concerning IFS theory with application to the geometric modelling and assignment of colours to IFS fractals and superfractals. The results will be described in the simplest setting of a single IFS with probabilities, but many generalizations are possible, most notably to superfractals.

Let the iterated function system (IFS) be denoted

$$\{\mathbb{X}; f_1, ..., f_N; p_1, ..., p_N\}. \tag{1.1}$$

This consists of a finite set of contraction mappings

$$f_n : \mathbb{X} \to \mathbb{X}, n = 1, 2, ..., N \tag{1.2}$$

acting on the compact metric space

$$(\mathbb{X}, d) \tag{1.3}$$

with metric d so that for some

$$0 \le l < 1 \tag{1.4}$$

we have

$$d(f_n(x), f_n(y)) \le l \cdot d(x, y) \tag{1.5}$$

for all $x, y \in \mathbb{X}$. The p_n's are strictly positive probabilities with

$$\sum_n p_n = 1. \tag{1.6}$$

The probability p_n is associated with the map f_n.

We begin by reviewing the two standard structures (**one** and **two**) that are associated with the IFS 1.1, namely its set attractor and its measure attractor, with emphasis on the *Collage Property*, described below. This property is of particular interest for geometrical modelling and computer graphics applications because it is the key to designing IFSs with attractor structures that model given inputs. Then we describe a new structure (**three**), the *fractal top* of an IFS.

Let

$$(\mathbb{H}(\mathbb{X}), h) \tag{1.7}$$

Date: December 10, 2003.

denote the compact metric space consisting of the compact non-empty subsets of $\mathbb{X}$, together with the Hausdorff metric. Define

$$F : \mathbb{H}(\mathbb{X}) \to \mathbb{H}(\mathbb{X}) \tag{1.8}$$

by

$$F(A) = \bigcup_n f_n(A) \tag{1.9}$$

Then F is a contraction mapping on $(\mathbb{H}(\mathbb{X}), h)$ with contractivity factor l and has a unique fixed point A such that

$$F(A) = A \tag{1.10}$$

and we have

$$\lim_{n\to\infty} A_n = A, \text{ where } A_{n+1} := F(A_n). \tag{1.11}$$

From Equation 1.10 A has the self-referential property

$$A = \bigcup_n f_n(A). \tag{1.12}$$

Equation 1.11 is the basis of the deterministic algorithm (e.g. in $\mathbb{R}^2$) where one starts from a set A_0 and computes a sequence of successive approximations $\{A_n\}$ to A. However, it is often more convenient to use the following Markov Chain Monte Carlo algorithm (MCMC or random iteration algorithm) as the basis of practical algorithms to compute approximations to A (for example when the f_n's are projective transformations acting on $\square$.) In this approach, we select an i.i.d. sequence of symbols

$$\sigma_1\sigma_2\sigma_3... \in \{1, 2, ..., N\}^\infty \tag{1.13}$$

with probabity p_n being associated with the choice $\sigma_k = n$. We also select $x_0 \in \mathbb{X}$ and let

$$x_{n+1} = f_{\sigma_{n+1}}(x_n).\text{for } n \geq 0. \tag{1.14}$$

Then almost always the limit set of $\{x_n\}$ is A. The proof is based on the ergodic theorem and the result is much more general. It provides in many case an efficient simple fast way to make pictures of A, and "zooms" on those pictures.

We are going to describe three structures that it is natural to associate with an IFS. Each structure describes or adds detail to A. Each structure may be computed by relatives of this random iteration algorithm.Of particular importance to modelling applications, each structure has what I call the Collage Property: namely

(1) the structure depends "continuously" on the parameters used to define it (coefficients of transformations, probabilities, ...);

(2) the structure has a self-referential property analagous to "A is made of N transforms of itself";

(3) the structure can be computed efficiently, given the maps and other parameters.

These properties enable interactive geometrical modelling of the structures. What do I mean here? Well we have for example the estimate

$$h(A, B) \leq \frac{h(B, F(B))}{1 - l}. \tag{1.15}$$

FIGURE 1. The point set attractor of an IFS of four projective transformations in $\mathbb{R}^2$. This illustrates the **first structure** associated with an IFS. It possesses the Collage Property and may be computed by random iteration.

So if we want to choose transformations so that the fixed point A is close to a target B, we simply have to choose maps so that

$$B \simeq \bigcup_n f_n(B) \tag{1.16}$$

In other words, we can start from B, the set we are trying to model, and choose maps based on their effect on B. Then the MCMC algorithm allows us to fine tune our approxation interactively.

EXAMPLE: This illustrates the **first structure** associated with an IFS, its point set attractor A. This possesses the Collage Property: the leafy structure of the attractor in Figure 1 is no accident. To create it I actually started from a leaf and used "Collage" software to find the maps. (**After the lecture and during drinks I will be delighted to demonstrate how this works in practice.) The following four projective transformations are used. In order from top to bottom these represent maps 1,2,3, and 4 respectively. Map #1 represents the main frond, Map#2 the stem, and Maps#3 and #4 represent the two lowest main fronds, on the left and the right at the bottom.

$$f(x,y) = \left(\frac{ax+by+c}{gx+hy+j}, \frac{dx+ey+f}{gx+hy+j}\right) \tag{1.17}$$

a	b	c	d	e	f	g	h	j
1.90080	-0.072	0.1857	0.01539	1.6914	0.0277	0.5626	-0.20119	2.0052
0.0020	-0.044	0.0755	0.0027	-0.044	0.104	0.0025	-0.0877	0.154
0.96543	-0.3519	0.05809	1.31389	-0.0651	-0.191	1.34823	-0.30675	0.0747
-0.3252	-0.0581	-0.0289	-1.229	-0.0011	0.1991	-1.2815	0.243	-0.0585

The **second structure** that it is natural to associate with an IFS, and which possesses the Collage Property, is the invariant measure of the MCMC process. The IFS with probabilities is associated with a contraction mapping

$$M : \mathbb{P}(\mathbb{X}) \to \mathbb{P}(\mathbb{X}) \tag{1.18}$$

FIGURE 2. This illustrates the **second structure** associated with an IFS. It is the invariant measure of the IFS, *with probabilities*. It too has the Collage Property. The IFS is the same as the one used in Figure 1.

where $(\mathbb{P}(\mathbb{X}), d_{MK})$ is the compact metric space consisting of the set of normalized Borel measures on $\mathbb{X}$ with the Monge-Kantorovitch metric, and where

$$M\mu = \sum_n p_n f_n \circ \mu \tag{1.19}$$

$(f \circ \mu)(\mathcal{B}) := \mu(f^{-1}(\mathcal{B}))$ for any Borel set $\mathcal{B} \subset \mathbb{X}$. This map too is a contraction and its fixed point can be thought of as a self-referential measure

$$\sum_n p_n f_n \circ \mu = \mu. \tag{1.20}$$

It can be computed by MCMC, and obeys the Collage estimate

$$d_{MK}(\mu, \mu_B) \leq \frac{d_{MK}(\mu_B, M\mu_B)}{1-l} \tag{1.21}$$

So the invariant measure of an IFS with probabilities has the Collage Property. In two dimensional examples as above one now starts from a greyscale image, its degrees of brightness representing the measure. Then one adjusts not only the maps, but also the probabilities, applied to the target image, to try and make the result look as much as possible like the target image, as in the following example.

EXAMPLE: The same maps are used as above. In fact this IFS consists of a "stem" map, with associated weight 0.05(approx.), a "main frond" map, with associated weight 1.8(approx.), and to lateral frond maps both with weights 0.5 (approx.). In Figure 2 the attractor is rendered in shades of green, with brightest green representing pixels with greatest measure. The probabilies are obtained by normalizing the weights. The image was computed using random iteration, and reveals far more structure (indeed, the new structure, the measure itself) than Figure 1.

The **third structure** is the *fractal top*. This too is naturally associated with the IFS, appears to possess the Collage Property, and provides a fascinating rich new object that provides new insights into the behaviour of attractors of IFS in the overlapping case and has immediate application (as will demonstrate) to low bandwidth digital content creation applications.

In the case of fractal tops we exploit the ordering of the maps implicit in the definition of an IFS. To illustrate, suppose that we have two maps with an overlapping attractor. Define a mapping (the "tops mapping")

$$T:\mathbb{H}(\Omega\times\mathbb{X})\rightarrow\mathbb{H}(\Omega\times\mathbb{X}) \tag{1.22}$$

where $\Omega=\{1,2,...,N\}^{\infty}$ denotes the code space asssociated with the IFS, equipped with the usual metric to make it into a compact metric space, as follows:

$$T(\Theta)=Top(T_1(\Theta),...,T_N(\Theta)) \tag{1.23}$$

for all $\Theta\in\mathbb{H}(\Omega\times\mathbb{X})$ where

$$T_n(\Theta)=\{T_n(\omega,x)=(n\omega,f_n(x)|(\omega,x)\in\Omega\times\mathbb{X}\} \tag{1.24}$$

where $n\omega$ means the concatenation of the string ω with n on the front. Clearly Ω has a natural lexigraphic ordering and all spaces involved are compact. In the above, for any pair of sets A and B in $\mathbb{H}(\Omega\times\mathbb{X})$ we define

$$Top(A,B)=\{(\omega,x)\in A\cup B|\omega\geqslant\varpi\text{ whenever }(\varpi,x)\in A\cup B\} \tag{1.25}$$

OBSERVATION: T possesses a unique fixed point, $\Upsilon\in\mathbb{H}(\Omega\times\mathbb{X})$. We call Υ the *fractal top* associated with the IFS.Υ possesses the Collage Property. In two dimensional case finite resolution approximations to Υ may be computed by a variant of the random iteration algorithm applied to the IFS

$$\{\Omega\times\mathbb{X};T_1(\omega,x),...,T_N(\omega,x);p_1,...,p_N\} \tag{1.26}$$

(keeping track of the highest value of ω encountered at each pixel, as random iteration proceeds). It is also observed that Υ can be computed deterministically by direct iteration of T; that is, for any $\Theta_0\in\Omega\times\mathbb{X}$,

$$T^{\circ n}(\Theta_0)\overset{weakly}{\rightarrow}\Upsilon \tag{1.27}$$

See Figure **. This has the benefit of avoiding wasted computation; a sequence of approximations to the fractal top can be computed directly. Finally, Υ appears to depend continuously on the maps of the IFS.and another image

2. Elementary Description of Colour Stealing

2.1. **Basic Idea.** Start with two iterated function systems (IFSs) and an input image. Run the random iteration algorithm, applying the same random choice to each IFS simultaneously. One of the IFSs produces a sequence of points that lie on the input image. The other IFS produces a sequence of points that are coloured according to the colour value of the other IFS, read off from the input picture.

2.2. **Example 1.** SET-UP: Start from two IFSs, (the DRAWING IFS and the COLOURING IFS,) and an INPUT IMAGE, a coloured Red Green Blue (RGB) image, such as a photograph of a natural scene. Here I choose the following IFSs. The first IFS corresponds to the image that I want to render. In this case it is a fern. It acts on a suitably positioned rectangle $\square_A\subset\mathbb{R}^2$. This DRAWING IFS is
(2.1)
DRAWING IFS:.. $\{\square_A;f_1,f_2,f_3,f_4;p_1=0.01,p_2=0.125,p_3=0.125,p_4=0.84\}$

FIGURE 3. This is the fractal top associated with the same IFS used in the other figures. This illustrates the **third structure** associated with an IFS. The color is assigned to the fractal top via color stealing.

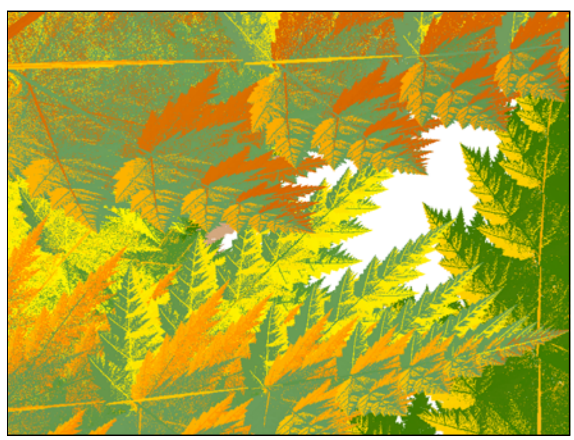

FIGURE 4. Close up on the fractal top in Figure 3. Details of the structure are revealed by colour-stealing.

where

$$\text{STEM:} \quad f_1(x,y) = (0.0x - 0.03y, 0.0x + 0.18y), \tag{2.2}$$

$$\text{LEFT FROND:} \quad f_2(x,y) = (0.2x - 0.2y, 0.2x + 0.2y + 0.1), \tag{2.3}$$

$$\text{RIGHT FROND:} \quad f_3(x,y) = (-0.2x + 0.2y, 0.2x + 0.2y + 0.1), \tag{2.4}$$

$$\text{MAIN FROND:} \quad f_4(x,y) = (0.85x + 0.0y, 0.0x + 0.85y + 0.2). \tag{2.5}$$

Recall that if the random iteration is applied using this IFS starting at the point $(x_0, y_0) = (0,0)$ (which lies upon the attractor), a sequence of points is generated $\{(x_n, y_n) : n = 1, 2, 3, ...\}$which lie upon the fern attractor. Normally these points are rendered according to the frequency with which corresponding pixels are visited (measure theory rendering- brighter colours for more frequently visited points...) or simply by marking the points in fixed colours, say green (set attractor rendering.) In the present case we have chosen the probabilities so that the IFS generates the attractor quite rapidly and uniformly. However we are going to colour the points according to the location of the output point of the COLOURING IFS.

The COLOURING IFS acts upon second rectangle $\square_B \subset \mathbb{R}^2$ that corresponds to the support of the INPUT IMAGE.

(2.6)
COLOURING IFS:.. $\{\square_B; g_1, g_2, g_3, g_4; p_1 = 0.01, p_2 = 0.125, p_3 = 0.125, p_4 = 0.84\}$

where

$$\text{RECTANGLE\#1:} \quad g_1(x, y) = (0.2x, 0.2y), \tag{2.7}$$

$$\text{RECTANGLE\#2:} \quad g_2(x, y) = (0.8x + 0.2, 0.2y), \tag{2.8}$$

$$\text{RECTANGLE\#3:} \quad g_3(x, y) = (0.2x, 0.8y + 0.2), \tag{2.9}$$

$$\text{RECTANGLE\#4:} \quad g_4(x, y) = (0.8x + 0.2, 0.8y + 0.2). \tag{2.10}$$

We start the random iteration of the second IFS at the point $(w_0, v_0) = (0, 0)$ and generate a sequence of points $\{(w_n, v_n) : n = 1, 2, 3, ...\}$ *by applying the same sequence of random choices* that we used for the DRAWING IFS. That is, the two IFS's are run in parallel, using only one sequence of random indices, selected according to the probabilities $p_{1,...,}p_4$. The point (x_n, y_n) is rendered in the colour of the point $(w_n, v_n) \in INPUT\ IMAGE$.

A beautiful coloured fern often appears on $\square_A$. See Figure 5 for a couple of examples.

I sometimes call this algorithm the code space colouring algorithm, for obvious reasons. The beauty of the images that it produces it comes in part from these facts (i) continuity of the map from codespace to fractal, which means that the map INPUT PICTURE to FRACTAL has a high degree of continuity; (ii) natural images have wonderful colour palettes inherently, with areas of varying types of continuity and special distributions of discontinuities, with colours that are naturally harmonious or complementary - these are stolen and wrapped seeming unrecognizably, often, but preserving many of their properties, around the fractal.

A few comments. Images vary smoothly if the input image is varied smoothly, leading to lovely animation. Images also vary smoothly when the COLOURING IFS is adjusted smoothly. There is no need to worry about the initial points, any values will do, as long as the two IFSs are run for a while before plotting begins. The example here of the fern has been chosen mainly for pedagogical reaons. Very powerful imaging systems are envisaged, using this algorithm to render complex and beautiful image sequences for content industry applications. One can think of this method as a variant of texture mapping, where the objective of the mapping is not a collection of polygons, but is instead a (collection of) fractals.

2.3. **Example 2.** Remarkable continuous image trasformations, that change fractal dimensions are obtained as in this this example, and many others in the same vein. Apply Example 1 except choose the following DRAWING IFS, which corresponds simply to a different tiling of the square by rectangles, from the one in the

FIGURE 5. Showing the same IFS rendered using two different INPUT IMAGES, parts of which are also in the picture.

COLOURING IFS:
(2.11)
DRAWING IFS:.. $\{\square_A; f_1, f_2, f_3, f_4; p_1 = 0.25, p_2 = 0.25, p_3 = 0.25, p_4 = 0.25\}$

where

$$\#1: \quad f_1(x,y) = (0.5x, 0.5y), \tag{2.12}$$

$$\#2: \quad f_2(x,y) = (0.5x + 0.5, 0.5y), \tag{2.13}$$

$$\#3: \quad f_3(x,y) = (0.5x, 0.5y + 0.5), \tag{2.14}$$

$$\#4: \quad f_4(x,y) = (0.5x + 0.5, 0.5y + 0.5). \tag{2.15}$$

In this case the output image is a continuous distortion of the INPUT IMAGE - it appears fast by random iteration. A pair of images corresponding to two such distortions is shown in Figure 6

FIGURE 6. Input image (left) and output image, using the code space colouring algorithm as in Example 2.

2.4. **Example 3.** The same technique can be applied to V-Variable fractals, but the implementation details are a bit more complicated, requiring several more image-sized buffers with various contents.(Another version of the algorithm based on random iteration to make V-variable fractals can be done also.) Here I describe the case N=Arbitrary, V=Arbitrary, M=4. The COLOURING IFS and the INPUT IMAGE are chosen the same as in the above two examples. The idea here is that the functions in the IFSs that constitute the superIFS are assumed to "correspond" as in this example: all of the functions $f_1^1, f_1^2, ...f_1^N$ correspond to the stem of the fern, all of the functions $f_2^1, f_2^2, ...f_2^N$ correspond to the left frond of the fern, ...and so on.... The to any point on any V-variable fractal we can associate a *colouring code* (I should mention here that I always use a "Z-buffer" as mentioned elsewhere, to resolve conflicts involving overlapping points) as illustrated here:

$$\text{POINT ON A V-VARIABLE FRACTAL: } f_3^1 \circ f_1^2 \circ f_2^4 \circ f_4^2 \circ f_4^N ... \tag{2.16}$$

corresponds to colouring code

$$31244... \tag{2.17}$$

and is coloured according to the colour of the point

$$g_3 \circ g_1 \circ g_2 \circ g_4 \circ g_4.... \tag{2.18}$$

in the INPUT IMAGE. The point is that corresponding parts of the successive V-variable fractals, produced by the random iteration algorithm, are coloured in the same way.,

2.5. **Example 4.** I should mention here that I think that there are many implications coming from this new class of algorithms, for image compression as well as computer graphics, that really need to be explored. For example, if one uses the SAME IFS for both the DRAWING IFS and the COLOURING IFS, but different resolutions on the INPUT IMAGE and the OUTPUT IMAGE, one obtains different resolutions for the two, but the way in which the resolution is changed can be very subtle.

2.6. **Applications.** Production of textures: start from one picture of a texture and make many other related ones. Valuable in Digital Content Production for computer games, animation, internet multimedia.

FIGURE 7. 2-variable fractal coloured with code space colouring algorithm.

FIGURE 8. Another 2-variable fractal (lying on the same superfractal) coloured by the code space colouring algorithm.

FIGURE 9. A 2-variable fractal coloured using one of the input images in Figure 1.

3. Detailed Description of Single IFS Random Iteration Colour Stealing.

Again, to help develop the main ideas, we give here a description of one way of implementing Colour Stealing with Priority Ordering using Random Iteration with a Single IFS, in contrast to the situation with a SuperIFS.

First we describe the image buffer that will correspond to the eventual *output image*. Let the *array* $\Box_A$ denote an image buffer, that is, an array of red, green, and blue pixel values, with an additional entry at each pixel, a certain real number p that is called the *priority* associated with the pixel. Thus, we write

(3.1)
$$\textit{(array)}\ \Box_A = (r_{m_A,n_A}, g_{m_A,n_A}, b_{m_A,n_A}, p_{m_A,n_A})\ m_A = 1,...,M_A,\ n_A = 1,...,N_A.$$

The positive integers M_A and N_A give the dimensions of the array. We also write $\Box_A(\subset \mathbb{R}^2)$ to denote a rectangular subset of the euclidean plane $\mathbb{R}^2$; this rectangle is partitioned into sub-rectangles indexed by the coordinates (m_A, n_A); each sub-rectangle corresponds to a pixel with colour components r_{m_A,n_A}, g_{m_A,n_A}, and b_{m_A,n_A}, and with an associated priority value. Let the discretization map

$$D_A : \Box_A(\subset \mathbb{R}^2) \to \textit{(array)}\ \Box_A \tag{3.2}$$

be defined by

$$D_A(a) = (m_A, n_A), \tag{3.3}$$

the index of the unique subrectangle of $\square_A(\subset \mathbb{R}^2)$ to which a belongs, for all $a \in \square_A(\subset \mathbb{R}^2)$.

Similarly we describe a second image buffer that corresponds to the *input image* from which colours will be *stolen,* (maybe a digital photograph of a painting by Cezanne, Renoir, or Seurat.) Let the *array* $\square_B$ denote an array of red, green, and blue pixel values,

(3.4)

$$\textit{(array)}\ \square_B = (r_{m_B,n_B}, g_{m_B,n_B}, b_{m_B,n_B}),\ m_B \in \{1,...,M_B\},\ n_B \in \{1,...,N_B\}.$$

These values are assumed to be given, or input, at the start of the algorithm. They will be used to determine the colour values of the array $\square_A$. There is no priority value in this case. The positive integers M_B and N_B give the dimensions of the array $\square_B$. As above write $\square_B \subset \mathbb{R}^2$ to denote a rectangular subset of the euclidean plane $\mathbb{R}^2$, partitioned into sub-rectangles indexed by the coordinates (m_B, n_B); each sub-rectangle corresponds to a pixel with colour components r_{m_B,n_B}, g_{m_B,n_B}, and b_{m_B,n_B}. Also, we define a discretization map

$$D_B : \square_B(\subset \mathbb{R}^2) \to \textit{(array)}\ \square_B \tag{3.5}$$

by

$$D_B(b) = (m_B, n_B), \tag{3.6}$$

the index of the unique subrectangle of $\square_B(\subset \mathbb{R}^2)$ to which b belongs, for all $b \in \square_B(\subset \mathbb{R}^2)$.

Now let there be given an IFS, to be called *drawing IFS,*

$$\{\square_A \subset \mathbb{R}^2; f_1, ..., f_N; p_1, ..., p_N\} \tag{3.7}$$

with attractor $A \subset \square_A \subset \mathbb{R}^2$. N is a positive integer and the p_ns are non-negative numbers summing to unity, representing *probabilities*. The functions f_n for all $n \in \{1, ..., N\}$ map from $\square_A \subset \mathbb{R}^2$ into itself and are such that the attractor A of the IFS is unique and can be computed using the standard random iteration algorithm. The probabilities are assumed to be such that when the standard random iteration algorithm is applied using the IFS, an efficient job is made of plotting or *drawing* the attractor A, at the resolution of the array $\square_A$.

And let there be given a second IFS, to be called the *colouring IFS*

$$\{\square_B \subset \mathbb{R}^2; g_1, ..., g_N; p_1, ..., p_N\} \tag{3.8}$$

with attractor $B \subset \square_B \subset \mathbb{R}^2$. The functions g_n for all $n \in \{1, ..., N\}$ map from $\square_B \subset \mathbb{R}^2$ into itself and are such that the attractor B of the IFS is unique and can be computed using the standard random iteration algorithm.

We need one more IFS, the *prioritizing* IFS,

$$\{[0,1] \subset \mathbb{R}; h_1, ..., h_N; p_1, ..., p_N\} \tag{3.9}$$

where

$$h_n(x) = p_1 + ... + p_{n-1} + \frac{\epsilon + 2xp_n}{2 + 2\epsilon} \tag{3.10}$$

for all $n = 1, ..., N$, for all $x \in [0,1]$, where $p_0 := 0$. This IFS is such that its attractor is totally disconnected, a cantor set contained in the real interval $[0,1]$. Let $D : [0,1] \to [0,1]$ be a discretization map.

The basic **colour stealing random iteration algorithm** proceeds as follows.

1. Initialize the image array $\square_A$ to zero, that is, set

$$r_{m_A,n_A} = g_{m_A,n_A} = b_{m_A,n_A} = 0 \text{ for all } m_A = 1,...,M_A,\ n_A = 1,...,N_A. \tag{3.11}$$

2. Initialize random iteration for the drawing, colouring, and prioritizing IFSs by selecting

$$a \in A, b \in B, x = 0. \tag{3.12}$$

3.Randomly select $\sigma \in \{1,...,N\}$ with $\sigma = n$ with probability p_n.
4. Calculate

$$newa = f_\sigma(a), newb = g_\sigma(b),\ \textit{and}\ newx = h_\sigma(x). \tag{3.13}$$

5. Calculate

$$(m_A, n_A) = D_A(newa),\ \textit{and}\ (m_B, n_B) = D_B(newb). \tag{3.14}$$

6. If $D(newx) > p_{m_A,n_A}$ then set

$$(r_{m_A,n_A}, g_{m_A,n_A}, b_{m_A,n_A}, p_{m_A,n_A}) = (r_{m_B,n_B}, g_{m_B,n_B}, b_{m_B,n_B}, D(newx)) \tag{3.15}$$

7. If sufficiently many iteration steps have been implemented, then output the digital image array $\square_A$ to storage, display and/or printer. Otherwise update $a, b,$ *and* x according to

$$a = newa, b = newb, and\ x = newx \tag{3.16}$$

and go to step 3.

3.1. **A Few Illustrative Images.** In this Section there are presented a few sample images, to give an advance taste of a few of the many image types that are discussed in this article.

Another picture:aThe following images show some other colour stealing images using very simple IFSs. At least a hint of the fascination and attractiveness of such images is provided.

4. Generalizations

It is natural to extend these notions to V-variable fractals, superIFS and superfractal to include the case of maps contractive on the average, more than a finite number of maps, more than a finite number of IFSs, IFSs with a variable number of maps, IFSs operating on sets which are not necessarily induced by point maps, other methods of constructing the probabilities for a superIFS, probabilities that are dependent upon position etc. But for reasons of simplicity and in order to illustrate key features we have not treated these generalizations at any length.

5. Acknowledgements

These notes describe an independent part of an ongoing collaboration with John Hutchinson at the Australian National University on Superfractals. This work has benefitted from his frequent discussions, his warm encouragement, and from the stimulating environment that he and his research group provide at the ANU.

FIGURE 10. This image was produced by random iteration with an IFS of four maps acting on $\mathbb{R}^2$, with a simple colouring IFS. Priority ordering, associated with the Theory of Fractal Tops, was used. This image can be made to "come alive", its colours and textures shimmering and changing in subtle and beautiful ways, by shifting the image from which the colours are stolen. This can be done in "real time". The graininess of this image is produced by cutting off computation before the image is fully rendered. Such graininess can be efficiently avoided by following some of the techniques described in this paper. See also Figures 11 and 12.

References

[1] M. F. Barnsley, *Fractals Everywhere*, Academic Press, New York, NY, 1988.

[2] M. F. Barnsley and S. Demko, *Iterated Function Systems and the Global Construction of Fractals,* R. Soc. Lond. Proc. Ser. A Math. Phys. Eng. Sci. 399(1985), pp. 243-275.

[3] M.F.Barnsley and L.F. Barnsley, *Fractal Transformations*, To appear.in a collection of articles about fractals, companion to the documentary film, “The Colours of Infinity” (GordonFilms 1995).

[4] M.F.Barnsley, J.E.Hutchinson, O. Stenflo *A Fractal Valued Random Iteration Algorithm and Fractal Hierarchy* (2003) submitted for publication.

[5] M.F.Barnsley, J.E.Hutchinson, O. Stenflo *V-Variable Fractals and Superfractals (2003)* submitted for publication.

[6] J. Elton, *An Ergodic Theorem for Iterated Maps,* Ergodic Theory Dynam. Systems, 7(1987), pp. 481-488.

[7] J. E. Hutchinson, *Fractals and Self-Similarity*, Indiana. Univ. Math. J., 30 (1981), pp. 713-749

FIGURE 11. Similar to Figure 10.

AUSTRALIAN NATIONAL UNIVERSITY, CANBERRA, ACT, AUSTALIA
E-mail address: Mbarnsley@aol.com

FIGURE 12. Similar to Figure 10.

FIGURE 13

Figure 14

Figure 15

Figure 16