5–10 July 2004, Antalya, Turkey — Dynamical Systems and Applications, Proceedings, pp. 706–711

# On Fractal Colouring Algorithms

Nergiz Yaz
Department of Mathematics, Faculty of Sciences
Ankara University, Tandoğan 06100 Ankara, TURKEY
E-mail: nergizyaz@yahoo.com
and
H. Hilmi Hacısalihoğlu

**Abstract**

Colouring algorithms define how fractals are coloured. The fractal formula creates the basic shape of the fractal, and colouring algorithms provide ways to colour that shape. This work is a survey on fractal colouring algorithms.

**Key words:** Colouring algorithms, Fractals.

## 1 Introduction

Fractals are virtually limitless computer graphics created by generator programs which run complex mathematical formulae. It is necessary to colour fractals in order to visualize them. The secret of their artistic dimension is the choice of colours. There are several generator programs available on the internet and most of them are freeware.

Fractals start with two steps which are choosing a method to create and then applying a colouring algorithm to draw. Mostly it starts with a process called *iteration*. An operation is repeated over and over again. Once the initial picture is produced, it is possible to zoom into different areas. Often a very simple formula gives rise to an incredibly complicated appearing image. Colouring algorithms within the generator make possible an almost infinite number of variations.

A fractal window represents a part of the complex plane, and when there is no transformation active each pixel has a unique complex number. On the first step, a fractal formula takes each of these complex numbers and runs it through iteration process. Then process produces a sequence of values. The positions of these points draws an *orbit*. In the second step, the user adjust the mapping of these numbers into colours. When it comes to colouring, there are mainly two choices: limited colour, full colour. If limited colour is chosen, then we have a colour *palette*, a series of colours from 1 to a finite number, $N$, say 256, each number can correspond to one colour on the list. In the case when the value of the number is greater than 256, then the generator program restarts with the first colour on the list.

So the beauty of fractal pictures relies on the choice of colours; limited or full. In this work we attempt to classify the most known colouring algorithms of fractals. All the figures appear here were created by UltraFractal generator program except Figure 3.

We start with the most classical and oldest colouring algorithms — The divergence algorithm.

## 2 The divergence algorithm

The divergence algorithm (with full colour) lies in the use of a scale of colours. It is based on the escape speed of each sequence from a chosen region since all of escaping sequences diverge to infinity. When you have a fractal formula, you run a point corresponding to the pixel through the formula. You choose a bailout value, for example, for a Mandelbrot or Julia sets, it is the circle corresponding to radius of 2. Then the simplest situation is that the pixels whose orbits never escape, *i.e.*, they are inside the bailout circle, are coloured black. The pixels that are outside are coloured in lots of different ways. Figure 1 is given by the basic black-white divergence algorithm.

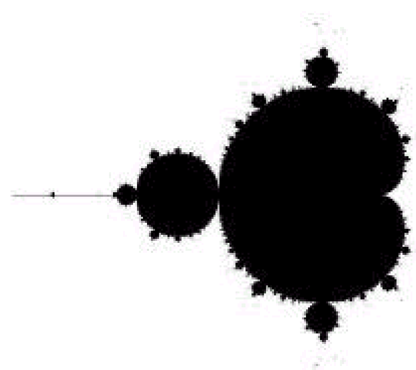

Figure 1

Then the following question arises, "After how many iterations does the orbit of a certain point tend to infinity?" The divergence algorithm colours the pixels in terms of how many iterations it takes to escape the bailout circle (Figure 2).

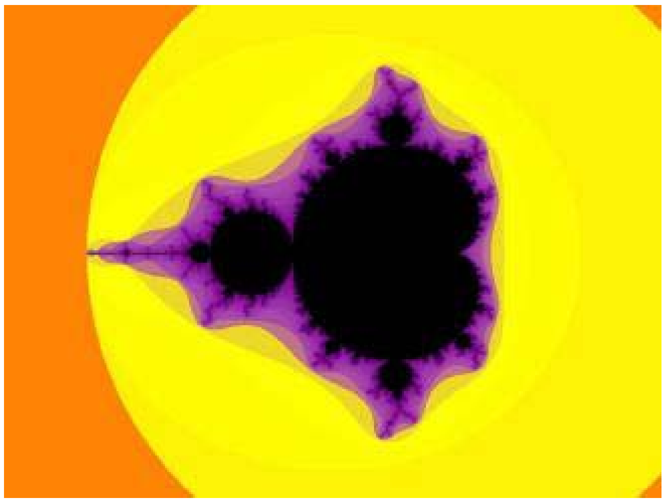

Figure 2

It is worthwhile to note that it is possible to choose regions other than a circle. They can be the shape of triangles, ellipses and so on. Also it is obvious that it is impossible to have a number of colours greater than the maximum number of iterations chosen for the calculation. So the divergence algorithm can be considered as a measurement of the distance from any point $z$ to the border of the set, but this distance is non-Euclidean. There is less common colouring algorithm of colouring a fractal image–distance estimation algorithm. This algorithm allows one to see every pixel that contains any points in the fractal set. It is based on the derivative of the iteration function.

## 3 The convergence algorithm

Some sets of orbit sequences do not tend to infinity. Among those sequences that stay within a finite range, many converge to a point in the set. Then it is possible to colour each of the points in the fractal set according to the speed of convergence of the corresponding sequence. This algorithm is given as $z$ converges to a fixed point, $|z_{n+1} - z_n|$ tends to zero. When the difference falls in some region, then the point is considered converged to the *attractor* and it is coloured. It could be coloured with any convenient algorithm.

This algorithm is always used to process fractal pictures that are created by *Newton-Raphson method*.

## 4 Decomposition algorithm

In this algorithm, $z$ is iterated as a point in the plane and its angle with $x$ axis is taken into the palette, *i.e.*, $z$ with angles above the real axis are given one colour, and $z$ with angles below the real axis are given another. Below is the *binary decomposition* algorithm of Julia set (Figure 3) [1].

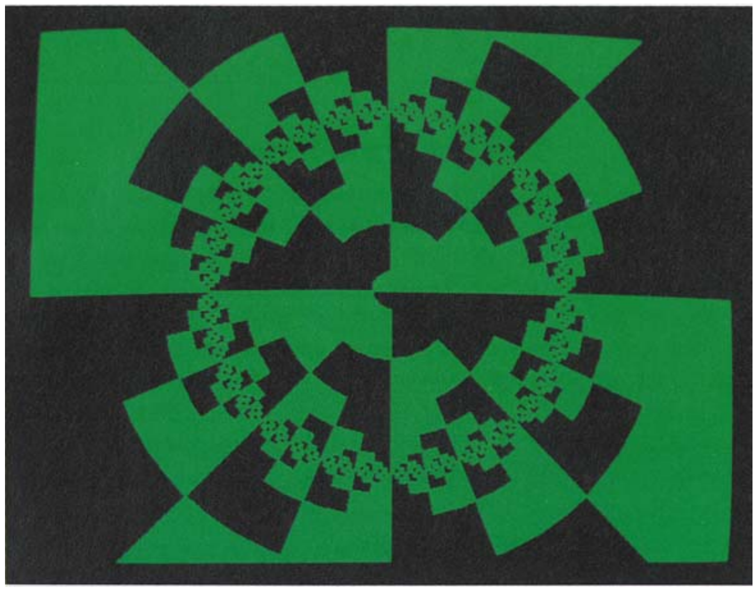

Figure 3

While this can provide some valuable sights into the structure of fractals, it is not often used in artistic fractal images.

The next colouring algorithm we consider is called *orbit traps*.

## 5 The orbit traps

This is the most used algorithm. It makes possible so many variations on colours. In this algorithm, a region on the complex plane is chosen, then the behavior of the $z$ values with respect to this region is investigated. The region can be a point, a line, a spiral, flowers, etc. The principle is to test whether the orbit of a certain point falls into the chosen region or not. If it falls into, then the point is taken as trapped. Then the iteration ends and the point is coloured based on the distance to the center of the chosen region.

In more mathematical terms, an orbit trap is a test which can be applied to complex numbers. Suppose we choose the region disk of radius 0.5 with its center at $(0.55, 0.15)$ for Mandelbrot Set's generating function $z \to z^2 + c$. Then we ask, "Is this complex number within 0.5 units of distance from the point $0.55 + 0.15i$ or not?". If the answer is yes for the point $z$ at any time during the iteration, then $z$ is taken as trapped by the orbit trap. The colouring is based on how far from the center of the orbit trap disk the point $z$ was (Figure 4, Figure 5, Figure 6, Figure 7).

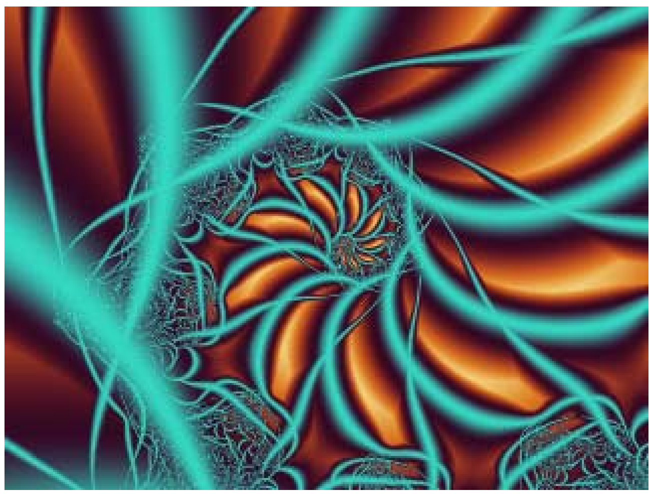

Figure 4

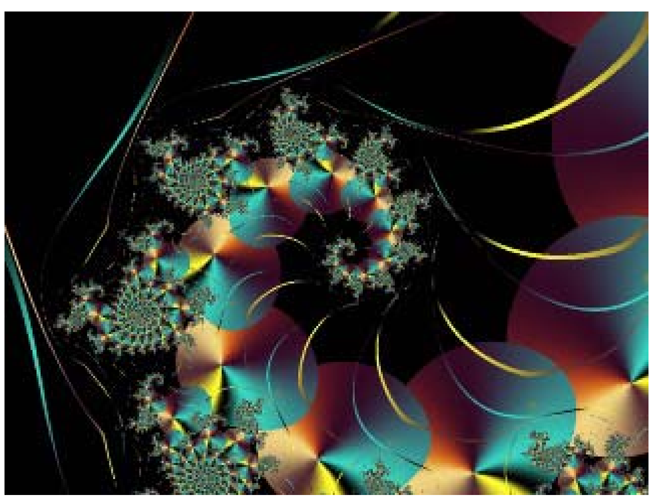

Figure 5

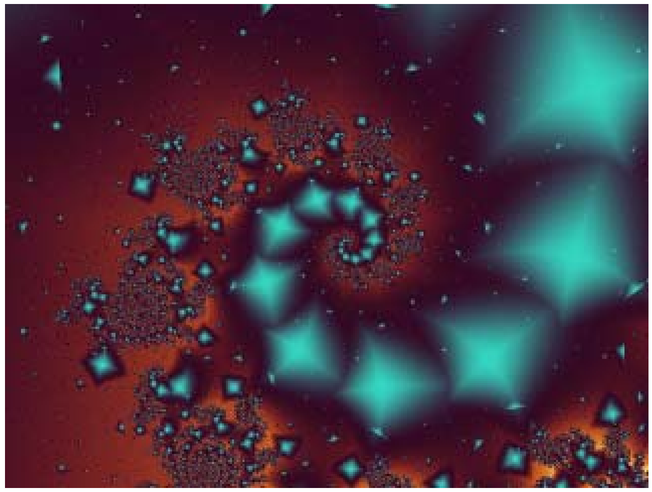

Figure 6

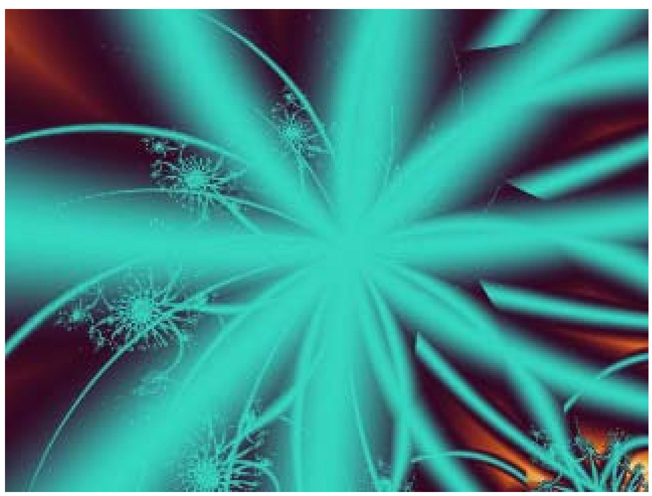

Figure 7

There are more variations on colouring such as colouring a point with different colours according to its distance, its magnitued or angle of $z$.

There are many other algorithms such as Biomorphs, Smurf, Stalks, ratios.

## References

[1] PEITGEN H.-O. AND SAUPE D., *The Science of Fractal Images*, Springer–Verlag, ISBN 0-387-96608-0.