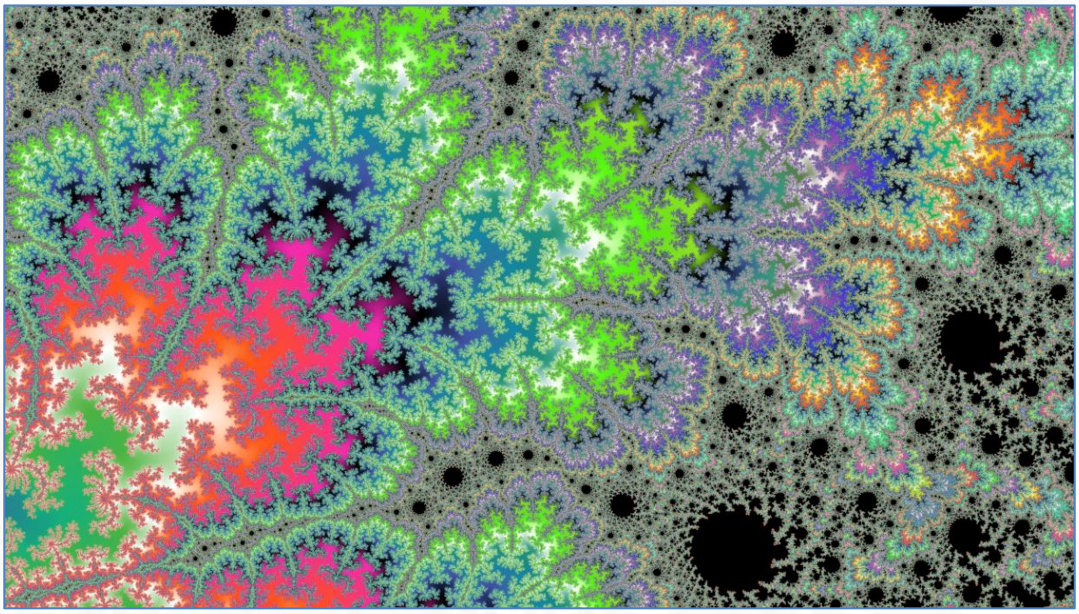
Extended Palette Algorithm

Sergio CT



« The mind that opens to a new idea never returns to its original size …»

Albert Einstein

https://fractalfun.es/en

Contents

Introduction

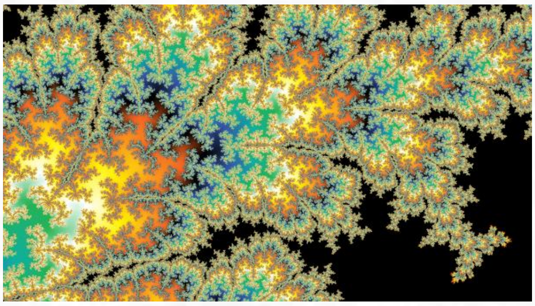
This document describes a method for dynamically extending a colour palette in such a way that the perceptible repetition of chromatic patterns, typical of fractals generated using escape-time algorithms, is eliminated. The goal is to maintain continuous, smooth and visually coherent colouring throughout the zooming process, even when the iteration value grows without bound.

This work was inspired by an idea originally proposed by **Christian Kleinhuis**, who kindly shared a preview of his formula prior to its release. During our conversation, he pointed out that there were no known implementations capable of addressing this problem in a fully satisfactory way, which motivated the development of this independent alternative algorithm.

For further information about Christian Kleinhuis' original proposal [illegible] reader is refer[illegible]

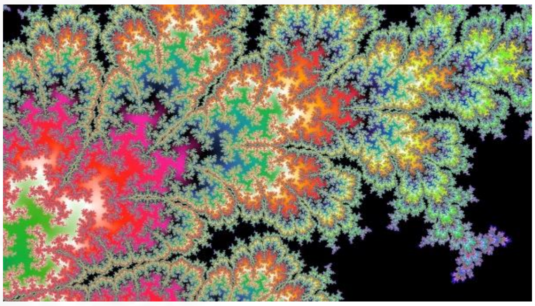
Motivation and objectives

The main objective of this work is to improve the visual quality of colouring in escape-time fractals by avoiding the appearance of repetitive chromatic patterns, even under very deep zooms. The intention is to generate a potentially unlimited set of colour variations that remain coherent with the base palette chosen by the user, while preserving smooth transitions free of perceptible discontinuities.



Escape time

→



Escape time with extended palette algorithm

To this end, a method is proposed that combines progressive hue rotations with carefully structured interpolations, so that the palette evolves naturally without losing its original identity.

Algorithm

The **extended palette algorithm** is based on the use of smoothed escape iteration and on the repeated application of the golden angle to the hue of the colours in the base palette. This combination makes it possible to generate a continuous, coherent and non-periodic chromatic evolution.

The golden angle applied to the hue is defined as:

$$\varphi_{hue} = 360° \cdot \left(1 - \frac{1}{\varphi}\right), \qquad \varphi = \frac{1 + \sqrt{5}}{2}$$

$$\varphi_{hue} \approx 137.50776405003785°$$

This value is incommensurable with 360°, meaning that there is no integer number of rotations that brings the hue back to its initial position. As a consequence, each successive rotation produces a distinct, non-repeating colour variation. This property is essential to avoid perceptible cycles during deep zoom explorations.

The main steps of the algorithm are described below.

- **Dynamic generation of the extended palette**

  Instead of constructing a fixed extended palette, colours are generated dynamically whenever they are needed. To do so, each colour in the base palette is transformed by applying a hue rotation proportional to the integer value $\boldsymbol{k}$ obtained from the smoothed iteration.

  Given a colour whose original hue is $\boldsymbol{H}$, the rotation is defined as:

$$H' = (H + k \cdot \varphi_{hue}) \bmod 360°$$

  where:

  - H is the original hue of the colour.
  - k comes from the decomposition of the smoothed iteration value (detailed in the following section).
  - $\varphi_{hue} \approx 137.50776405003785°$ is the golden angle applied to the hue.

  As a consequence of this cumulative rotation, a non-periodic and potentially unlimited sequence of related chromatic versions is obtained.

- **Colour assignment using smoothed iteration**

For each point of the fractal, a smoothed escape iteration value $\boldsymbol{v}$ is computed. Since the base palette contains $\boldsymbol{L}$ colours, $\boldsymbol{v}$ is normalised by dividing it by $\boldsymbol{L}$, so that it can be interpreted in blocks of that size.

$$\frac{\boldsymbol{v}}{\boldsymbol{L}} = \boldsymbol{k} + \boldsymbol{f}$$

where:

- $k = \left\lfloor \frac{v}{L} \right\rfloor$ is the integer part of v and determines the accumulated hue rotation applied to the base colours.
- $f = \frac{v}{L} - k$ is the fractional part of v, indicating the continuous position within the block and controlling the progressive transition between the base palette and its rotated version.

The combined use of $\boldsymbol{k}$ and $\boldsymbol{f}$ allows the colouring to evolve without jumps

- $k \in \mathbb{Z}$ determines how many successive golden-angle rotations are applied.
- $f \in [0,1)$ regulates the smooth blending within each palette block.

From this decomposition, the final colour is obtained by following the steps described below.

1. Selection of the pair of colours from the base palette

The continuous position within the block, represented by $\boldsymbol{f}$, determines which two consecutive colours from the base palette will take part in the final interpolation. To obtain this position within the palette, the following is computed:

$$\boldsymbol{x} = \boldsymbol{f} \cdot \boldsymbol{L}$$

where $\boldsymbol{L}$ is the number of colours in the palette.

From $\boldsymbol{x}$ we obtain:

$$\boldsymbol{i} = \lfloor \boldsymbol{x} \rfloor$$

$$\boldsymbol{\alpha} = \boldsymbol{x} - \boldsymbol{i}$$

where:

- i is the integer part of x and determines the index of the lower colour of the pair.
- α is the fractional part of x and indicates the blending proportion towards the next colour $i + 1$.

In this way, the two consecutive colours from the base palette to be interpolated are defined, together with the exact fraction by which they must be blended.

It is important not to confuse this parameter $\boldsymbol{\alpha}$, used to interpolate between these two consecutive colours, with the value $\boldsymbol{f}$, which will later be used to blend each base colour with its corresponding rotated version.

2. Dynamic generation of the rotated colour

Once the two base colours $\boldsymbol{B_i}$ and $\boldsymbol{B_{i+1}}$, have been selected, their extended versions are obtained by applying a hue rotation proportional to the previously computed integer value $\boldsymbol{k}$. The accumulated rotation is given by:

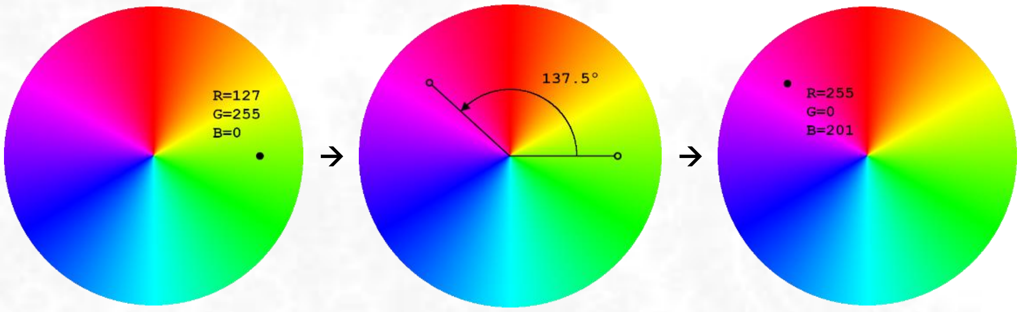
$$\boldsymbol{\theta_k = k \cdot \varphi_{hue}}$$

Each base colour is then transformed as follows:

1. The colour is converted from RGB to HSL.
2. Its hue is rotated according to the transformation:

$$\boldsymbol{H' = (H + \theta_k)\, mod\, 360^\circ}$$

3. The colour is converted back to RGB, yielding the rotated colours $\boldsymbol{E_i}$ and $\boldsymbol{E_{i+1}}$.



Original colour Hue rotation Rotated colour

This cumulative rotation generates a non-periodic sequence of colour variations, avoiding visible repetitions even after multiple increments of $\boldsymbol{k}$.

3. **Blending between base and extended colours**

Each base colour $\boldsymbol{B_i}$ and $\boldsymbol{B_{i+1}}$ is combined with its corresponding rotated version $\boldsymbol{E_i}$ and $\boldsymbol{E_{i+1}}$. This blending is controlled by the fractional value $\boldsymbol{f}$, which determines the degree of transition between the original palette and the extended palette within each block.

The resulting blends are defined as:

$$\boldsymbol{M_i = (1 - f)\, B_i + f\, E_i}$$
$$\boldsymbol{M_{i+1} = (1 - f)\, B_{i+1} + f\, E_{i+1}}$$

where:

- M_i is the interpolated version of the colour with index i,
- M_{i+1} is the interpolated version of the colour with index $i + 1$.

These values $\boldsymbol{M_i}$ and $\boldsymbol{M_{i+1}}$ constitute the two colours between which the final interpolation will be performed in the next step of the algorithm.

4. **Final interpolation between the two blended colours**

The fractional part $\boldsymbol{\alpha}$, computed earlier, determines the exact position between the two already blended colours $\boldsymbol{M_i}$ and $\boldsymbol{M_{i+1}}$. The final colour is obtained by means of a linear interpolation between them:

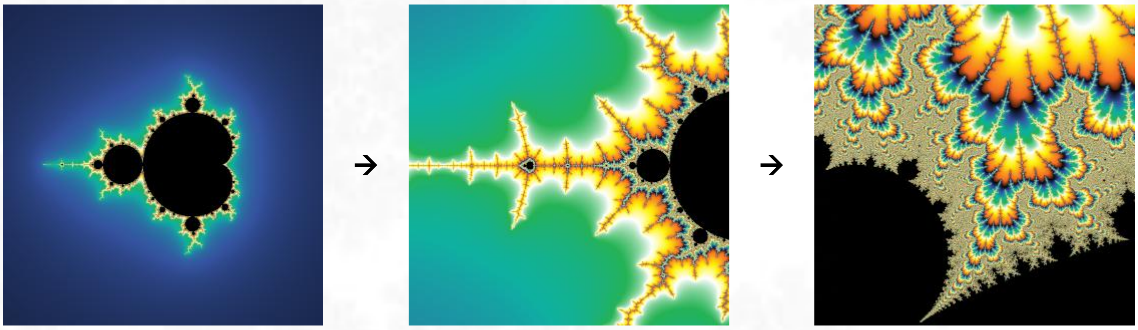
$$\boldsymbol{C(v) = (1 - \alpha)\, M_i \;+\; \alpha\, M_{i+1}}$$

This operation provides a smooth and continuous transition as one moves through the palette block, ensuring that no perceptible jumps or discontinuities occur. The result $\boldsymbol{C(v)}$ is the final colour assigned to the fractal point.

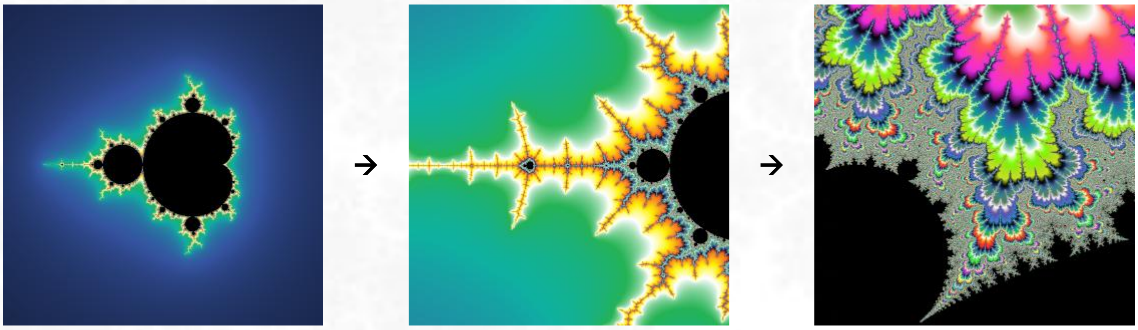
Thanks to this construction, the colour evolves smoothly with respect to v, the transition between chromatic blocks remains continuous, and the rotation based on the golden angle prevents the appearance of repetitive patterns even during very deep zooms.

Conclusions

The method described makes it possible to extend the original palette in a continuous and non-periodic manner without the need to construct a fixed extended palette. Since the rotation applied to the hue is proportional to the integer value $\boldsymbol{k}$, when $\boldsymbol{k} = \mathbf{0}$ the rotation is null and the extended colours coincide exactly with the base colours. This implies that the initial iterations of the fractal use the palette chosen by the user without alteration, ensuring that the initial colours faithfully reflect that selection. From that point onwards, the transition towards more complex chromatic variations occurs gradually, smoothly and coherently as the smoothed iteration value increases.



Zoom sequence with fixed palette



Zoom sequence with extended palette

The algorithm is modular and flexible, making it easy to experiment with variants such as non-linear interpolations, dynamic rotations, blending in other colour spaces or user-defined gradients, while always maintaining smooth and coherent transitions even during deep zoom explorations.