School of Computing
FACULTY OF ENGINEERING

UNIVERSITY OF LEEDS

Signed Distance Fields as a Means of Improving the Speed of Rendering Three-Dimensional Fractals

Eleanor Joan Mills

Submitted in accordance with the requirements for the degree of MSc High Performance Graphics and Games Engineering

2021-2022

The candidate confirms that the following have been submitted.

Items	Format	Recipient(s) and Date
Project Report	Report	SSO (19/08/22)
Project Code	Zip File	SSO (19/08/22)

Type of project: Exploratory Software

The candidate confirms that the work submitted is their own and the appropriate credit has been given where reference has been made to the work of others.

I understand that failure to attribute material which is obtained from another source may be considered as plagiarism.

(Signature of Student) Nell Mills

Summary

<Concise statement of the problem you intended to solve and main achievements (no more than one A4 page)>

Acknowledgements

<The page should contain any acknowledgements to those who have assisted with your work. Where you have worked as part of a team, you should, where appropriate, reference to any contribution made by other to the project.>

Note that it is not acceptable to solicit assistance on 'proof reading' which is defined as the "the systematic checking and identification of errors in spelling, punctuation, grammar and sentence construction, formatting and layout in the test";

see http://www.leeds.ac.uk/gat/documents/policy/Proof-reading-policy.pdf.

Contents

Chapter 1

Introduction

Chapter 2

Background

This project aims to improve the efficiency of real-time rendering of 3D fractals. This chapter will focus on background theory for fractals, signed distance functions and sphere tracing.

2.1 2D Fractals - The Mandelbrot Set

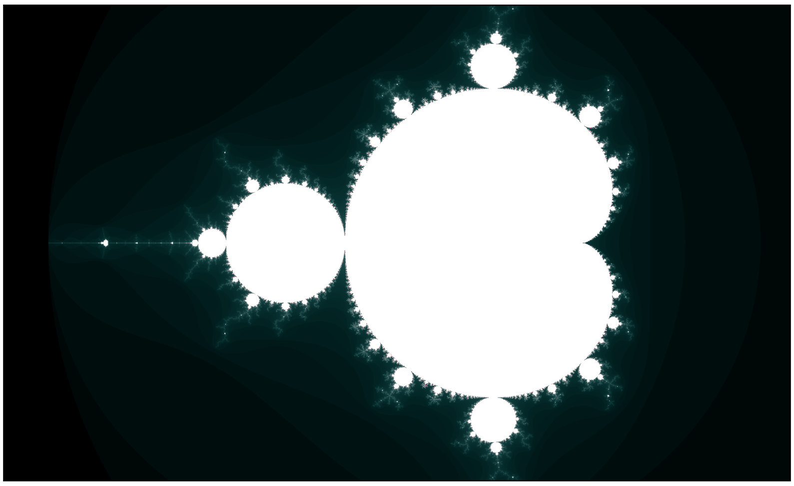

Figure 2.1: The Mandelbrot set. The white points in the centre are inside the set.

The Mandelbrot set is the set of two-dimensional points that satisfy a certain constraint on the following complex quadratic equation:

$$Z = Z^2 + C \tag{2.1}$$

where Z and C are complex numbers. The constraint on the points is that their orbit must be bounded. The value of Z is initialized to 0 and equation 2.1 is iterated over, each new value of Z being placed back in to the equation in the next iteration. If the length of the point Z does not exceed a threshold, then the point (represented by C) is in the Mandelbrot set [1].

Figure 2.1 shows a generated Mandelbrot set. The real part of the point C is represented by the x-axis, and the imaginary part by the y-axis. Equation 2.1 is iterated over a maximum of five hundred times, and the threshold value is two. The pixels are coloured according to how many iterations are achieved before the length of Z exceeds the threshold.

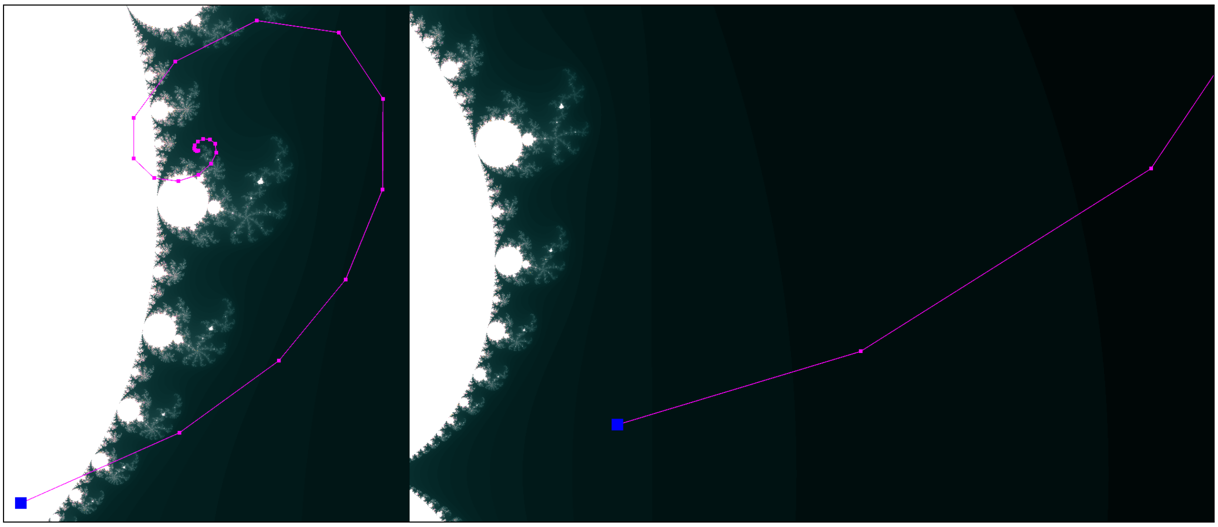

Figure 2.2: Visualization of the first twenty five iterations of equation 2.1 on the initial points [0.3, 0.05] (left) and [0.5, 0.04] (right). The initial points are shown in blue.

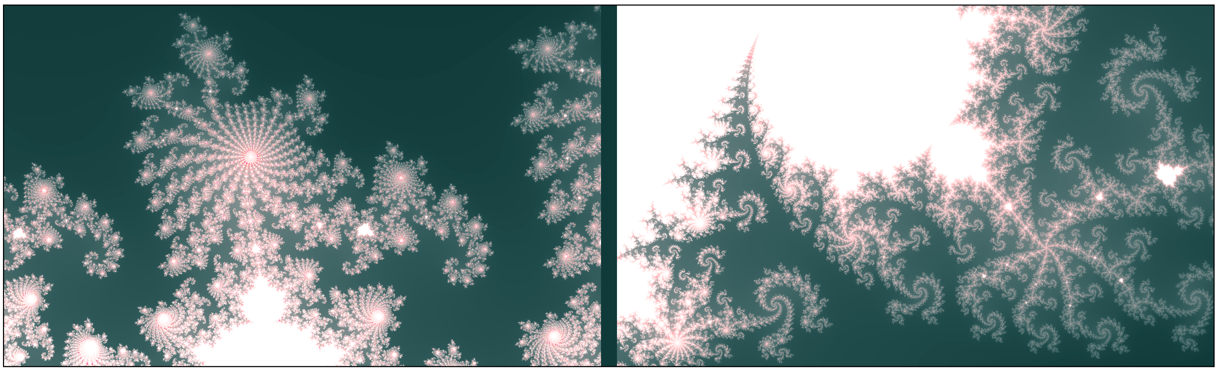

Figure 2.3: Two different views of the Mandelbrot set, zoomed in.

Figure 2.2 illustrates the first twenty five iterations on two different points. For the first point, the iterations converge in a spiral shape and the length of Z never exceeds the threshold of two, therefore the point is in the Mandelbrot set and is coloured white. For the second point, the iterations diverge and exceed the threshold of two within a few iterations, so this point is not in the Mandelbrot set and is coloured dark.

Figure 2.3 shows two zoomed-in views at the edge of the original shape. New patterns can be seen, as well as repeated ones, and even new instances of the original shape. This is because the Mandelbrot set has infinite detail, so if one decreases the range of the axes, new patterns will emerge [2].

This project makes use of three-dimensional fractal rendering, so the next section will look at the challenge of bringing this infinite level of detail to three dimensions.

2.2 3D Fractals - The Mandelbulb

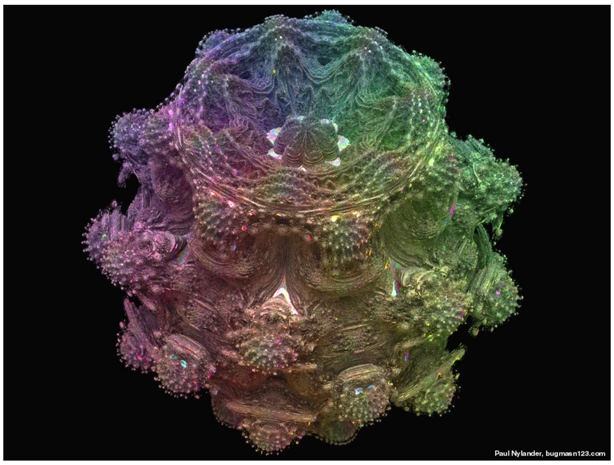

Figure 2.4: First look at the Mandelbulb, from Paul Nylander's website [3].

Since the Mandelbrot set is in two dimensions, and since complex numbers have only two components (real and imaginary), obtaining a true three-dimensional Mandelbrot set is a challenging task, and a mathematically rigorous three-dimensional Mandelbrot has not yet been found [4].

One attempt that seems to have come close was originated by Rudy Rucker in 1987. Rucker thought that expressing the three-dimensional points in spherical coordinates would allow the manipulation of the points in a similar way to the complex-number operations performed on the points in the two-dimensional Mandelbrot set [5].

Rucker did not have the computational power to accomplish a rendering of this idea, so it was put aside for twenty years, until Daniel White independently published a formula in 2007, which took the approach proposed by Rucker. White decided to approach the problem by considering the geometrical consequences of multiplying numbers in the complex plane, which amounts to rotating them [4].

White's formula produced images that looked promising, but they didn't have the level of detail that was expected from a true three-dimensional equivalent of the Mandelbrot set. A mathematician, Paul Nylander, raised White's formula to a higher power (eight), which would be equivalent to increasing the number of rotations of the point. The resulting image is shown in figure 2.4. The shape maintains excellent detail, even at high levels of magnification [4].

The new shape, known as the Mandelbulb, has roughly the same formula as the Mandelbrot set (equation 2.1):

$$Z = Z^k + C \tag{2.2}$$

where Z is raised to an arbitrary power like so:

$$Z^k = r^k(sin[k\theta]cos[k\phi], sin[k\theta]sin[k\phi], cos[k\theta]). \tag{2.3}$$

The variable r is the norm of Z (|Z|), θ is equal to $\arctan(Z_y/Z_x)$ and ϕ is equal to $|(Z_x, Z_y)|/Z_z$. The spherical coordinates of the point Z/|Z| are represented by θ and ϕ. Equations 2.2 and 2.3 are sourced from Chapter 33 of the book Ray Tracing Gems II [6].

2.3 Signed Distance Functions and Sphere Tracing

This section will focus on rendering three-dimensional fractal shapes, specifically the Mandelbulb. Firstly, signed distance functions will be explained, followed by a very brief explanation of ray tracing. Finally, the application of the signed distance functions to ray tracing will be discussed.

2.3.1 Signed Distance Functions

Signed distance functions provide an estimate of how close a point is to the surface of a shape. If the result of the function is positive, then the point is outside the surface. If the result is negative, then the point is inside the surface. If the result is zero, then of course the point is exactly on the surface of the shape described by the function [7].

A signed distance function can be derived using the Böttcher map for the fractal formula, which is a deformation of the space. Closer to the surface of the fractal, the space is deformed to a greater degree than parts further away from the surface, as shown in figure 2.5. The deformations of the space occur in such a way as to map the exterior of the fractal to the exterior of a unit disk [8].

A rigorous mathematical explanation of the Böttcher map will not be given in this paper, but a brief introduction is helpful to understand the derivation of the distance function for the Mandelbulb. The map can be calculated as follows:

$$\phi_C[Z] = lim_{n\to\infty}[f^n[Z]]^{k^{-n}} \tag{2.4}$$

where the value of f[Z] is the same as in equation 2.2 and n refers to the current iteration of the equation [6].

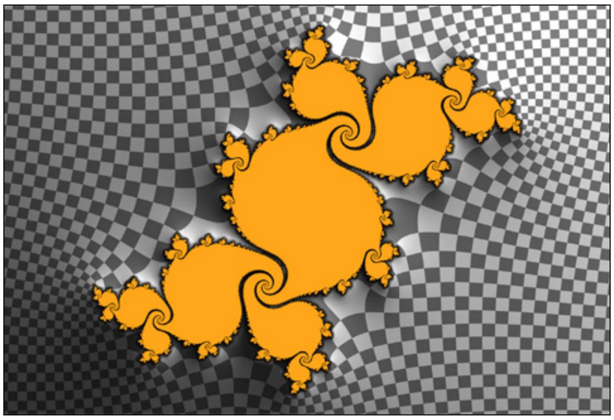

Figure 2.5: The Böttcher map generated for the Julia set (another two-dimensional complex fractal), from the website of Inigo Quilez [8].

Looking at equation 2.4, take a point Z that is not in the fractal, so that the function $f^n[Z]$ grows as the number of iterations increases, tending towards infinity. For a large enough value of n, the term $f_n[Z]$ will become vastly larger than the value of C (as the point is far from the fractal), meaning that C can reasonably be discarded [6].

From this casting away of C we obtain:

$$f^{n+1}[Z] \approx (f^n[Z])^k \tag{2.5}$$

for points that are sufficiently far away from the surface of the fractal. Since the point at iteration n is far from the fractal, if one were to undo all of the iterations to get back to the original point at $f_0[Z]$, we would obtain this expression:

$$f_0^n[Z]^{k^{-n}} = Z \tag{2.6}$$

and by extension, returning to the Böttcher map:

$$\phi_0[Z] = Z \tag{2.7}$$

which is the approximate result that equation 2.4 gives when the function $f^n[Z]$ ultimately diverges [6].

Next, we will use something called the Hubbard-Douady potential, equal to the logarithm of the modulo of the Böttcher map. This is a map of points onto a unit disk. Recall that the Böttcher map maps the exterior of the fractal to the exterior of a unit disk. This now becomes important, as it enables us to use the Hubbard-Douady potential for all of our complex fractals. We now have a function that tends towards zero as the points approach the boundary of the fractal [8]:

$$G[Z] = lim_{n\to\infty} \frac{log|f^n[Z]|}{k^n}. \tag{2.8}$$

Equation 2.8 is not ready to be used as a distance measurement yet. However,

2.3.2 Ray and Sphere Tracing

Ray tracing is a technique used for rendering scenes. Generally, a ray is a line that is cast from the camera or eye through each pixel of an image. Tests are performed to see which object (if any) is encountered by the ray first. By bouncing the ray from object to object, the program can gather the various light contributions from objects that reflect, emit or refract light [9].

Distance estimators can be used for ray tracing.

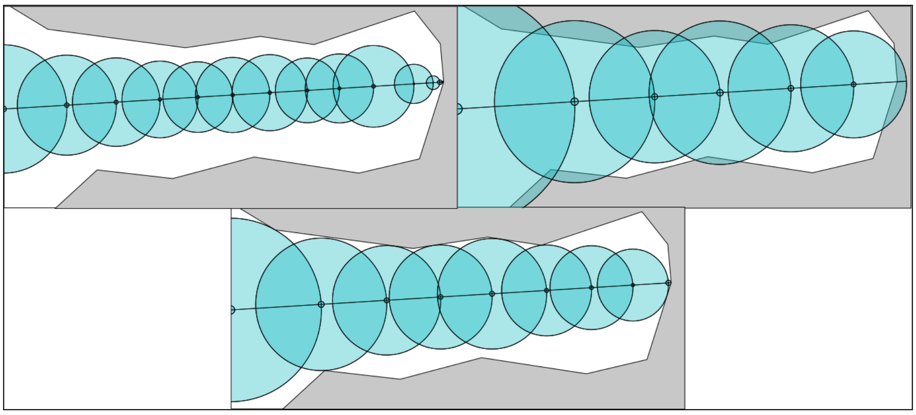

Figure 2.6: Three scenarios in sphere tracing. Top left: Distance function underestimates distance, resulting in a loss of performance. Top right: Distance function overestimates distance, resulting in a loss of accuracy. Bottom: Distance function is exact.

Papers:

- Ray Tracing Gems: High-Quality and Real-Time Rendering with DXR and Other APIs [9].
- Ray Tracing Gems II [6].
- Procedural Rendering w/Ray Marching [7].
- Distance to Fractals [8].

2.4 Signed Distance Fields

Papers:

- Signed distance fields: A natural representation for both mapping and planning [10].

- Stored values from signed distance function, stored as 2D texture or 3D grid.
- Saves having to recompute SDF all the time.
- Fractal scenes with high view distance could benefit from some pre-calculated values to give them a head start - show room of pillars and remark on frame rate (no SDF).
- Remark on memory costs.
- Briefly talk about storage methods - octrees.
- Talk about 2D methods - particularly used for fonts.

Chapter 3

Implementation

3.1 Project Structure and Overview

3.2 Rendering 3D Fractals

3.3 Signed Distance Field

3.4 Second Attempt: 2D Texture SDF

3.4.1 Single-pixel Sampling

3.4.2 Sampling a small area

3.4.3 Sampling with memory

3.5 Performance Measurement

3.5.1 Data Types Collected

3.5.2 Representative Views

3.5.3 Animation

Chapter 4

Results

4.1 3D SDF

4.1.1 Mandelbulb

4.1.2 Hall of Pillars

4.2 2D SDF

4.2.1 Mandelbulb

4.2.2 Hall of Pillars

Show four images:

- Top-Left: No SDF, no movement.
- Bottom-Left: No SDF, movement.
- Top-Right: 2D SDF, no movement.
- Bottom-Right: 2D SDF, movement.

Use these to demonstrate that the performance benefits only exist for still images generally.
Also, have an image of the 2D SDF with movement and no checks for changes in distance travelled - show artefacts and performance. Explain that if you want a performance boost all the time, you need to put up with these artefacts.
In "further work", maybe could say that it would be great to find a different method of removing artefacts without removing performance benefits of SDF. Mention that you tried sampling from an area but it was wobbly, and mention that adding a small decay didn't work either because of the difference in depth between the closer and farther parts of the image, and the performance benefit comes from skipping the edges anyway. Would have to find a way of detecting these artefacts and removing them.
Include results separately for having the second timestamp before and after the image copying. Remark that the shader execution time is significantly smaller but that the copying offsets that. Also investigate why on earth the framerate is so much better, but the timestamps don't reflect this...

Chapter 5

Conclusion

Things to mention:

- Ambient occlusion and how reducing the number of iterations ruins it.

Further work:

- Figure out a way to remove artefacts from movement without resetting distance for everything. Artefact detection.
- Find a way to copy the 2D SDF image faster.
- Make the 3D SDF move with the camera so that it can handle movement through the scene.
- Move the generation of the 3D SDF to the shaders, maybe make it coarser so it can be regenerated every frame (so it can handle animation)?

References

[1] R. L. Devaney, "The mandelbrot set, the farey tree, and the fibonacci sequence," *The American Mathematical Monthly*, vol. 106, no. 4, pp. 289–302, 1999.

[2] D. Ashlock, "Evolutionary exploration of the mandelbrot set," in *2006 IEEE International Conference on Evolutionary Computation*, pp. 2079–2086, IEEE, 2006.

[3] P. Nylander, "Hypercomplex fractals." `http://www.bugman123.com/Hypercomplex`, 2009. accessed: 31/07/22.

[4] J. Aron, "The mandelbulb: first 'true'3d image of famous fractal," *New Scientist*, vol. 204, no. 3736, pp. 54–55, 2009.

[5] R. Rucker, "In search of a beautiful 3d mandelbrot set." `https://www.rudyrucker.com/blog/notebooks/rucker_mandelbulb_ver7_sept24_2009.pdf`, 2009. accessed: 31/07/22.

[6] A. Marrs, P. Shirley, and I. Wald, *Ray Tracing Gems II: Next Generation Real-Time Rendering with DXR, Vulkan, and OptiX*. Springer Nature, 2021.

[7] C. Robles, H. Lee, S. Yin, and V. Dhar, "Procedural rendering w/ray marching,"

[8] I. Quilez, "Distance to fractals." `https://iquilezles.org/articles/distancefractals`, 2004. accessed: 5/08/22.

[9] E. Haines and T. Akenine-Möller, *Ray Tracing Gems: High-Quality and Real-Time Rendering with DXR and Other APIs*. Springer, 2019.

[10] H. Oleynikova, A. Millane, Z. Taylor, E. Galceran, J. Nieto, and R. Siegwart, "Signed distance fields: A natural representation for both mapping and planning," in *RSS 2016 workshop: geometry and beyond-representations, physics, and scene understanding for robotics*, University of Michigan, 2016.

Appendices

Appendix A

External Material

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Appendix B

Ethical Issues Addressed