

**Members :**

Nguyễn Tuấn Anh

Nguyễn Đình Niên

Đỗ Đức Kiên

Kiều Duy Đông

K55CA

Spring 2013

MANDELBROT SET

computer graphics ASsignment

Version 1.0

May 8, 2013

Contents

[I. INTRODUCTION 2](#_Toc355944312)

[1. Mandelbrot set 2](#_Toc355944313)

[2. 3D Mandelbrot set 2](#_Toc355944314)

[II. ALL WORK 4](#_Toc355944315)

[1. 2D Mandelbrot set implementation 4](#_Toc355944316)

[2. 3D Mandelbrot set implementation 4](#_Toc355944317)

[**a. Mathematics :** 4](#_Toc355944318)

[**b. How to draw Mandelbrot set 3D** 5](#_Toc355944319)

[C. **Using opengl to draw :** 5](#_Toc355944320)

[3. Conclusion 7](#_Toc355944321)

[III. FUTURE WORK 7](#_Toc355944322)

[IV. PLAN 8](#_Toc355944323)

[V. REFERENCE 9](#_Toc355944324)

# I. INTRODUCTION

## 1. Mandelbrot set

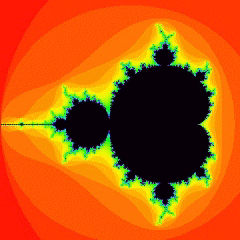


Figure : Mandelbrot 2d

The Mandelbrot set is a mathematical set of points whose boundary is a distinctive and easily recognizable two-dimensional fractal shape. The set is named after the mathematician Benoit Mandelbrot, who studied and popularized it.

Images of the Mandelbrot set are made by sampling complex numbers and determining for each whether the result tends towards infinity when a particular mathematical operation is iterated on it. Treating the real and imaginary parts of each number as image coordinates, pixels are colored according to how rapidly the sequence diverges, if at all.

More precisely, the Mandelbrot set is the set of values of c in the complex plane for which the orbit of 0 under iteration of the complex quadratic polynomial *zn*+1 = *zn*2 + *c* remains bounded.

Images of the Mandelbrot set display an elaborate boundary that reveals progressively ever-finer recursive detail at increasing magnifications. The "style" of this repeating detail depends on the region of the set being examined. The set's boundary also incorporates smaller versions of the main shape, so the fractal property of self-similarity applies to the entire set, and not just to its parts.

The Mandelbrot set has become popular outside mathematics both for its aesthetic appeal and as an example of a complex structure arising from the application of simple rules, and is one of the best-known examples of mathematical visualization.

## 2. 3D Mandelbrot set

The fractal calculation follows a similar process as a normal Mandelbrot set using the same formula ,w' = wn + c, but instead of using standard complex numbers w and c are hyper-complex **'triplex'** numbers with three components corresponding to the Cartesian x, y, and z co-ordinates.

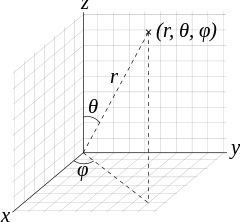


Figure 2:Spherical coordinates (r, θ, φ) as commonly used in physics: radial distance r, polar angle θ (theta), and azimuthal angle φ (phi). The symbol ρ (rho) is often used instead of r.

w = {x, y, z}n = rn { sin(θn) cos(φn), sin(θn) sin(φn), cos(θn) }

**where**:  
r = sqrt( x2 + y2 + z2 )  
θ = atan2( sqrt( x2 + y2), z )φ = atan2( y, x )

Equation : The triplex number w is raised to a power n

For each pixel a ray is stepped into the scene by a small amount. The x, y and z co-ordinates of the ray at this point provide the input triplex number for the fractal equation, which is then iterated until the magnitude of the triplex number exceeds a bailout value (usually 4.0), or the maximum iteration count is reached.

At the end of the iteration loop a distance estimation function is used to calculate the closest point in any direction to the fractal surface. It is defined as:

distance estimation = 0.5 \* |w| \* log(|w|) / |δw|

**where** |w| is the magnitude of the triplex number w and δw is the derivative.

**Note:** There are two method to calculate δw

* The jacobian distance estimator
* The scalar distance estimator (we will use it for our project)

*drn=n|fn−1(c)|(n-1)drn−1+1*

Equation : distance estimator formula

The distance estimation value is crucial to the ray tracing process. It tells us the maximum step distance the ray can move before we need to recalculate the fractal at the new location, which is far more efficient than a fixed step ray marching approach.

# II. ALL WORK

## 1. 2D Mandelbrot set implementation

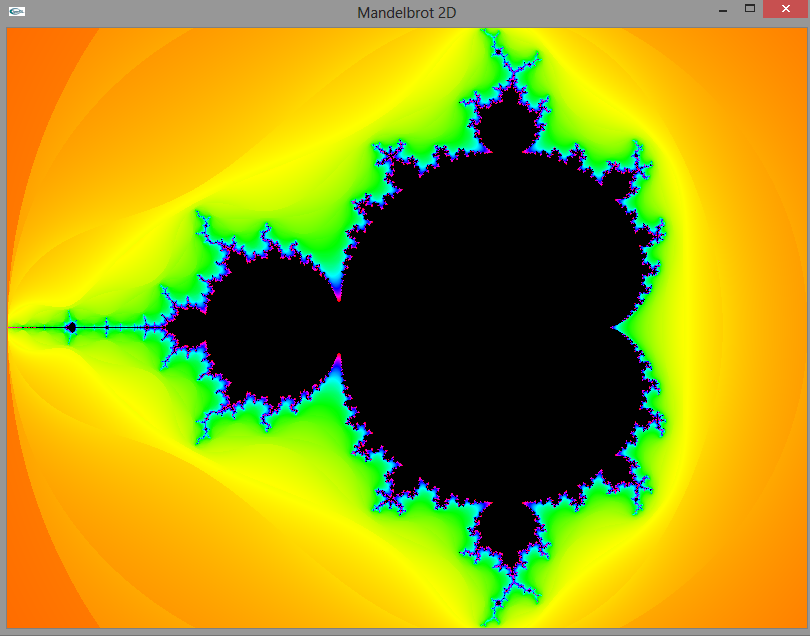


Figure : With formula z' = z2 +c, Iterations = 32

* Renders Mandelbrot 2D
* Color all point which are outside Mandelbrot set depend on **distance estimator**

<http://fraktal.republika.pl/cpp_sdl.html>

* Zoom in to show more details

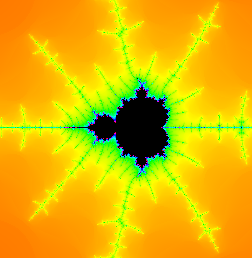


Figure : after zooming

* We program this application in project Dev C++ with library OpenGL

## 2. 3D Mandelbrot set implementation

### **a. Mathematics :**

**In search of the ideal 3D Mandelbrot set, we approach from ideal : limit the search to formulas that satisfy |w| = |{x, y, z}n| = (x2 + y2 + z2)n/2 and the challenge is to find the rotation matrix that produces the best looking results.**

**We have these definitions :**

**,**

**,**

r = sqrt( x2 + y2 + z2 )  
θ = atan2( sqrt( x2 + y2), z )φ = atan2( y, x )

Table :SOME METHODS ARE USED IN OUR PROJECT

**Example** : Classic method

**{x, y, z}n = Rz(θ). Ry(φ). {rn, 0, 0}**

**= rn { cos(θn) cos(φn), cos(θn) sin(φn), - sin(θn) }**

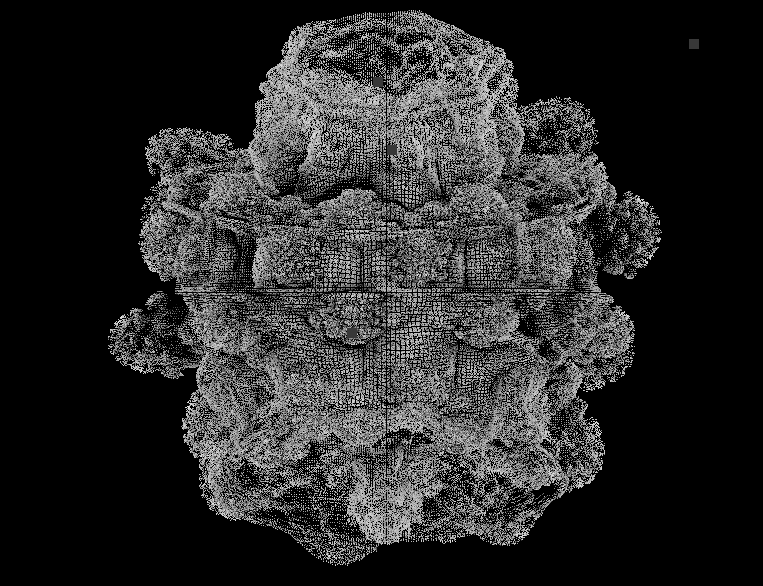
|  |  |  |
| --- | --- | --- |
| method | formula | note |
| Classic | **{x, y, z}n = rn { cos(θn) cos(φn), cos(θn) sin(φn), - sin(θn) }** |  |
| Positive z-component | **{x, y, z}n= rn { cos(θn) cos(φn), sin(θn) cos(φn), sin(θn) }** |  |
| Cosine | **{x, y, z}n= rn { sin(θn) cos(φn), sin(θn) sin(φn), cos(θn) }** |  |
| Rudy Rucker | **{x, y, z}n = rn { cos(θn) cos(φn), cos(θn) sin(φn), sin(θn) }** | **θ = tan-1(z/x)** |
| Unknown | **{x, y, z}n = rn { cos(θn) cos(φn), sin(θn) cos(φn), cos(θn) sin(φn)}** |  |

### **b. How to draw Mandelbrot set 3D**

* **Generating points set in 3D Mandelbrot set from formula**
* **Colorize points**
* **Camera views**
  + **camera rotation: change the rotation of the camera on each axis.**
  + **camera zoom: magnifies the current view. Note, to reveal more details on the surface the camera position has to be moved, you can't just zoom in.**
* **Passing parameters:**
  + **Power: the power n used in the fractal equation**
  + **maxIterations: higher values will reveal more detail when close to the fractal surface but increases calculation time.**
  + **bailout: the 'escape' threshold for the fractal iteration. A lower value tends to give rounder simplified surfaces.**

### **C. Using opengl to draw :**

Figure 5: FORMULA : Z' = Z8 +C, RANGE X,Y, Z = 400, ITERATIONS = 10



* Using C++ and OpenGL
* Using C#, OpenTK (include OpenGL APIs) and Windows Forms with visual passing parameters and increasing speed calculation

|  |  |  |  |
| --- | --- | --- | --- |
| Method | | Max iterations = 2 | Max iterations = 8 |
| Classic | |  |  |
| Positive z-component | |  |  |
| Cosine | |  |  |
| Rudy Rucker |  | |  |
| Unknown |  | |  |

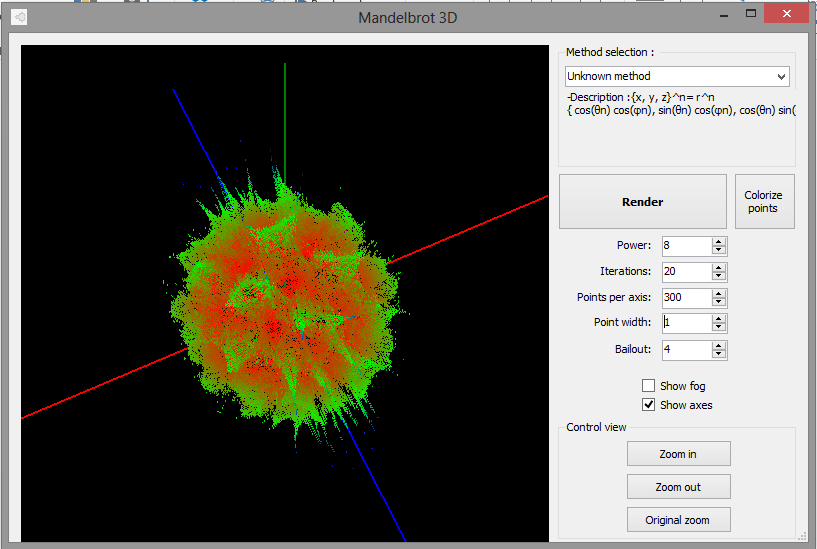


Figure : windows form draw 3D mandelbrot set

## 3. Conclusion

- Our project only plots point in Mandelbrot set, sometime they are quite ugly and non-travelling. Currently, on some forum such as <http://www.fractalforums.com/> , they rendered 3d Mandelbrot set with surface and travelling. Look them so amazing !

- About speed of processing, although we using multi-thread in C# windows forms, but speed still is slow. Solution for this is taking advantage of GPU. We can using some programming language on GPU such as CUDA from NVIDA

- On internet, there are some software generating 3D Mandelbrot set : Mandelbulb3D from <http://www.fractalforums.com/> , Mandelbulber of Krysztof Marczak,… All of them performed quite well.

# III. FUTURE WORK

* Render 3D Mandelbrot set travelling
* Using ray tracing to make set more beautiful
* Implement a program on GPU

# IV. PLAN

1. **Period**

This project is started from April 11st to May 10th about 6 weeks

1. **Main plan**

Table : Assign task

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task | Tuan Anh | Nien | Kien | Dong |
| Mathematics overall | roughly | roughly | roughly | All work |
| Mandelbrot 2D project Dev C++ | generate points | Colorize |  |  |
| Mandelbrot 3D project Dev C++ | lighting, texture, perspective view |  | generate points, colorize |  |
| Formulas Mandelbrot set |  |  | All formulas |  |
| Windows form C# |  | UI, generate points, colorize | Multithreads |  |
| Documentation | 2D Mandelbrot | 3D Mandelbrot | 3D Mandelbrot | Mathematics, introduction |

Table : Schedule tasks

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Task** | **1st week** | **2nd week** | **3rd week** | **4th week** | **5th week** | **6th week** |
| **Mathematics overall** |  |  |  |  |  |  |
| **Mandelbrot 2D project Dev C++** |  |  |  |  |  |  |
| **Mandelbrot 3D project Dev C++** |  |  |  |  |  |  |
| **Formulas Mandelbrot set** |  |  |  |  |  |  |
| **Windows form C#** |  |  |  |  |  |  |
| **Documentation** |  |  |  |  |  |  |

1. **Comment**

***“All member have got much enthusiasm for their work . In 6 weeks, there are some hard problems in technical such as : formula, colorize, … or non-technical such as : assignments from other subjects, … and now, this project is completed almost required features and we got many valuable experiences.***

***In team, Dong is 4-year student so he doesn’t have much time to take part in team’s project. However, he always is following the processing of project and distributes some his experiences for 3 members remains.” - Nien***

# V. REFERENCE

[1] . <http://www.fractalforums.com/theory/> - Theory about Mandelbrot set

[2]. <http://en.wikipedia.org/wiki/Mandelbrot_set> - Overview about Mandelbrot set

[3]. <http://www.skytopia.com/project/fractal/mandelbulb.html>

[4]. [http://blog.hvidtfeldts.net](http://blog.hvidtfeldts.net/) – More in distance estimator of 3D Mandelbrot set