Mandelbrot

Computer Graphics Assignment

Group 9 K56CA•5/21/2014

Instructor: Dr. Bui The Duy

Team Members:
Tran DucMuoi
Nguyen ThanhToan
Phan Dang Thanh

Mandelbrot

Computer Graphics Assignment

Instructor Dr. Bui The Duy

Team members

Tran Duc Muoi (Leader)	11020204	K56CA
Nguyen Thanh Toan	11020342	K56CA
Phan Dang Thanh	11020429	K56CA

Abstract

The objective of this report is giving both overview and discussion about Computer Graphics Assignment of group 9 called “Mandelbrot 3D”. In this report, we will discuss about Mandelbrot set and its 3D version. How we constructed and implemented is also mentioned. We divided the assignment into 2 main parts: Mandelbrot and Mandelbulb. In 6 weeks from 31 March, we researched about mentioned topics and implemented it step by step. The assignment ends at 21 May 2014.

Acknowledgment

This work is supported much by Dr. Bui The Duy and his PhD student, Ngo ThiDuyen. We also want to give special thanks to K55CA group for their report in 2013 that we use as reference and sample document.

Table of Contents

Project Description

Requirements

1. Draw Mandelbrot set 3D.
2. Function and color parameters are changeable.
3. Custom Camera View

Mandelbrot set

Introduction

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.

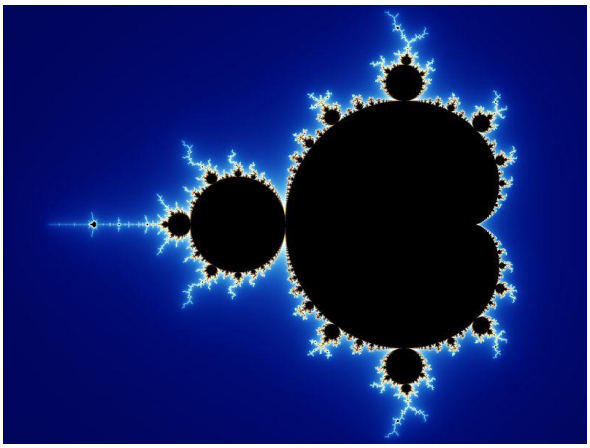

Figure 1 - Mandelbrot set

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.

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.

Formula

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 $z_{n+1} = z_n^2 + c$ remains bounded.

Mandelbulb set

Introduction

A canonical 3-dimensional Mandelbrot set does not exist, since there is no 3-dimensional analogue of the 2-dimensional space of complex numbers. It is possible to construct Mandelbrot sets in 4 dimensions using quaternions. However, this set does not exhibit detail at all scales like the 2D Mandelbrot set does. Note that, in this assignment, we only consider 3D one.

The Mandelbulb is a three-dimensional analogue of the Mandelbrot set, constructed by Daniel White and Paul Nylander using spherical coordinates.

The fractal calculation follows a similar process as a normal Mandelbrot set using the same formula, $w' = w^n + c$, 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.

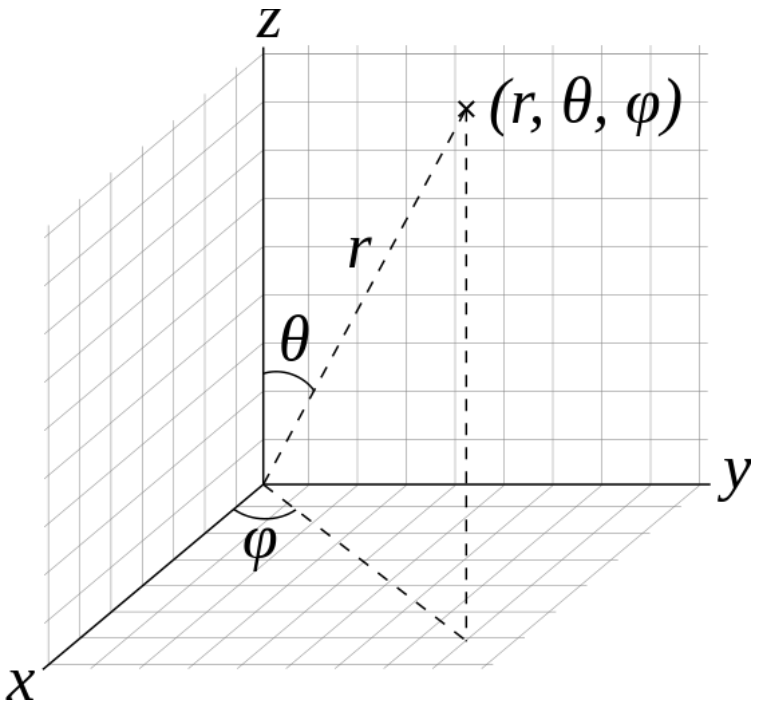

Figure 2 - Cartesian x, y, and z coordinates

The exponentiation term can be defined by:

$$w = \{x, y, z\}^n = r^n \{sin(\theta n)cos(\varphi n), sin(\theta n)sin(\varphi n), cos(\theta n)\}$$
$$r = sqrt(x^2 + y^2 + z^2)$$
$$\theta = atan2(sqrt(x^2 + y^2), z)$$
$$\varphi = atan2(y, x)$$

Implementation and Discussion

Mandelbrot Implementation

In this assignment, we are not asked to implement Mandelbrot, aka Mandelbrot 2D. However, we would like to better understand the algorithm in implementation, so we develop this part in Dev C++ and OpenGL.

Algorithms

We used the simplest algorithm for generating a representation of the Mandelbrot set is known as the escape time algorithm. A repeating calculation is performed for each x, y point in the plot area and based on the behavior of that calculation, a color is chosen for that pixel.

The x and y locations of each point are used as starting values in a repeating. The result of each iteration is used as the starting values for the next. The values are checked during each iteration to see if they have reached a 'bailout'. If that condition is reached, the calculation is stopped, the pixel is drawn, and the next x, y point is examined. For some starting values, escape occurs quickly, after only a small number of iterations. For starting values very close to but not in the set, it may take hundreds or thousands of iterations to escape. For values within the Mandelbrot set, escape will never occur. The programmer or user must choose how much iteration, or 'depth,' they wish to examine. The higher the maximum number of iterations, the more detail and subtlety emerge in the final image, but the longer time it will take to calculate the fractal image.

Escape Conditions

In this project, it's simply if the magnitude of a point is over bailout (we use 4.0 in this part) because the property of magnitude of Mandelbrot element is always less than 2.

Coloring

The color of each point represents how quickly the values reached the escape point. Black is used to show values that fail to escape before the iteration limit or elements of Mandelbrot set. For other pixels, the coloring algorithm is using gray level to indicate the iteration that the pixel escapes. This gives a visual representation of how many cycles were required before reaching the escape condition.

Tool and Libraries

Dev C++ and OpenGL

Mandelbulb Implementation

Distance Estimation

The Mandelbulb fractal is ray traced using basically the same process as the 4D Quarternion Julia set fractal: 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 which usually is 4.0 but in our project, 2.0 is chosen, 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:

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

$$where\ |w|\ is\ the\ magnitude\ of\ the\ triplex\ number\ w\ and\ \delta w\ is\ the\ derivative.$$

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.

Distance Estimator Method

There are a number of methods to calculate the distance estimator we mentioned before: Jacobean Distance Estimator, The Scalar Distance Estimator and The Escape Length Gradient Approximation. In this project, we use the second method.

$$dr_n = 8|f_{n-1}(c)|_7 dr_{n-1} + 1$$

where 'dr' is a scalar function and the order of Mandelbulb set is 8.

Camera Views

The camera views are available as follow:

1. Camera Rotation: Change the rotation of camera on each axis
2. Camera Zoom: Magnifies the current view
3. Camera Yaw
4. Camera Pitch

Coloring

We have been trying to figure out how to render my 3D Fractals for a while now and cannot seem to make a good looking image. Right now for 3D we are using a very basic distance estimation mixed with r, g, and b color space:

$$Color = colorValue * (|x|, |y|, |z|)$$

Where colorValue is ranged between from 0.0 to 1.0 means from the darkest to the brightest.

Passing Parameter

As a requirement of the assignment, we construct the following parameters that are able to pass.

1. nOrder: This is the order of 3D Mandelbulb, means the power n used in fractal equations.
2. iteration: The higher its value is, the more detail of the fractal but more time consuming.
3. colorValue: Range between 0.0 – 1.0. The higher its value is, the brighter color will be.

Tool and Libraries

We used Eclipse IDE to implement this part. However, base on real experiment, we turned into Visual Studio 2010 because its time consuming was a significant down. OpenGL was also used in this part.

Result

The following are the results of our assignment.

Mandelbrot

Note that iteration = 32

Initialization

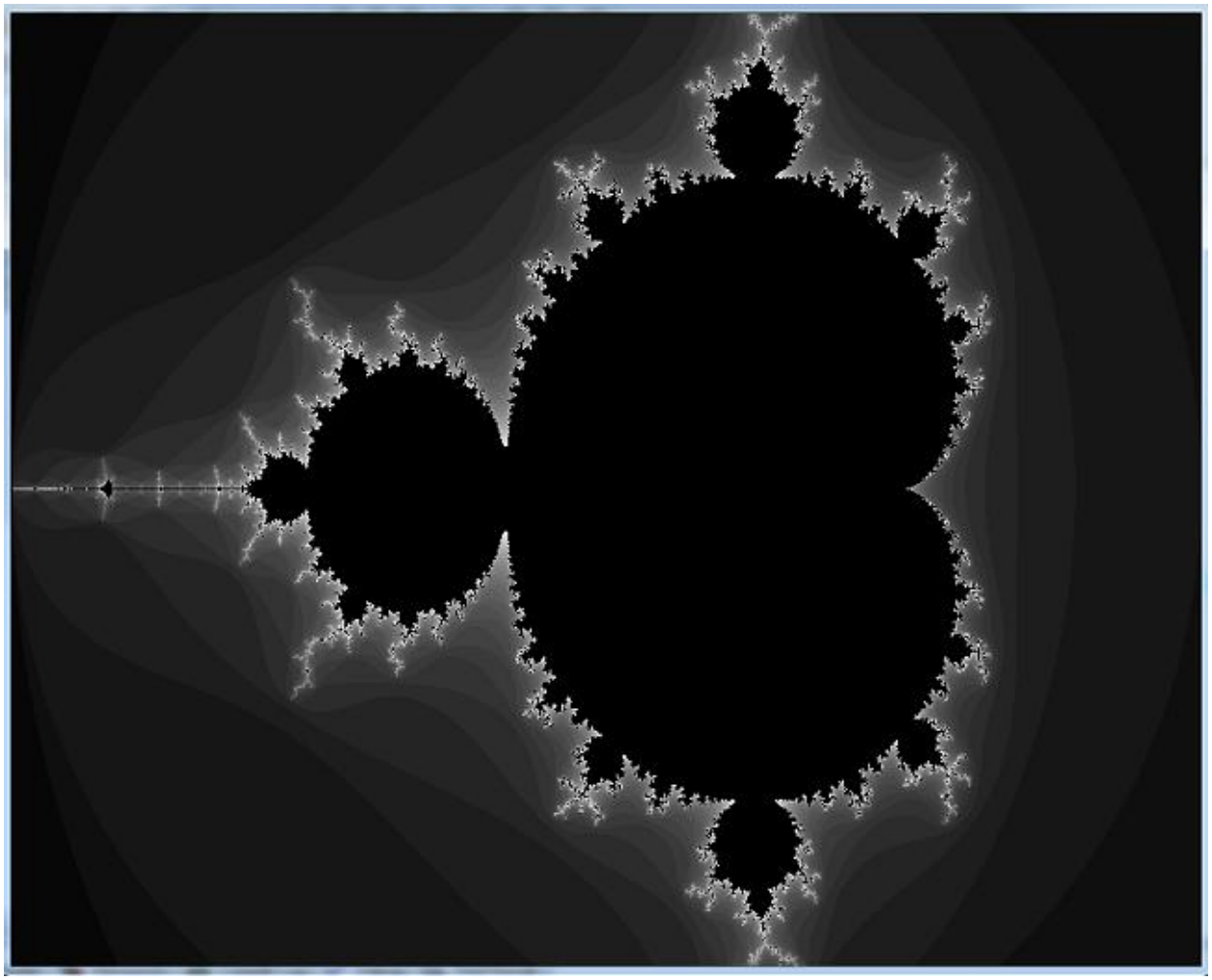

Figure 3 - Initial Mandelbrot with smooth gray

After zooming

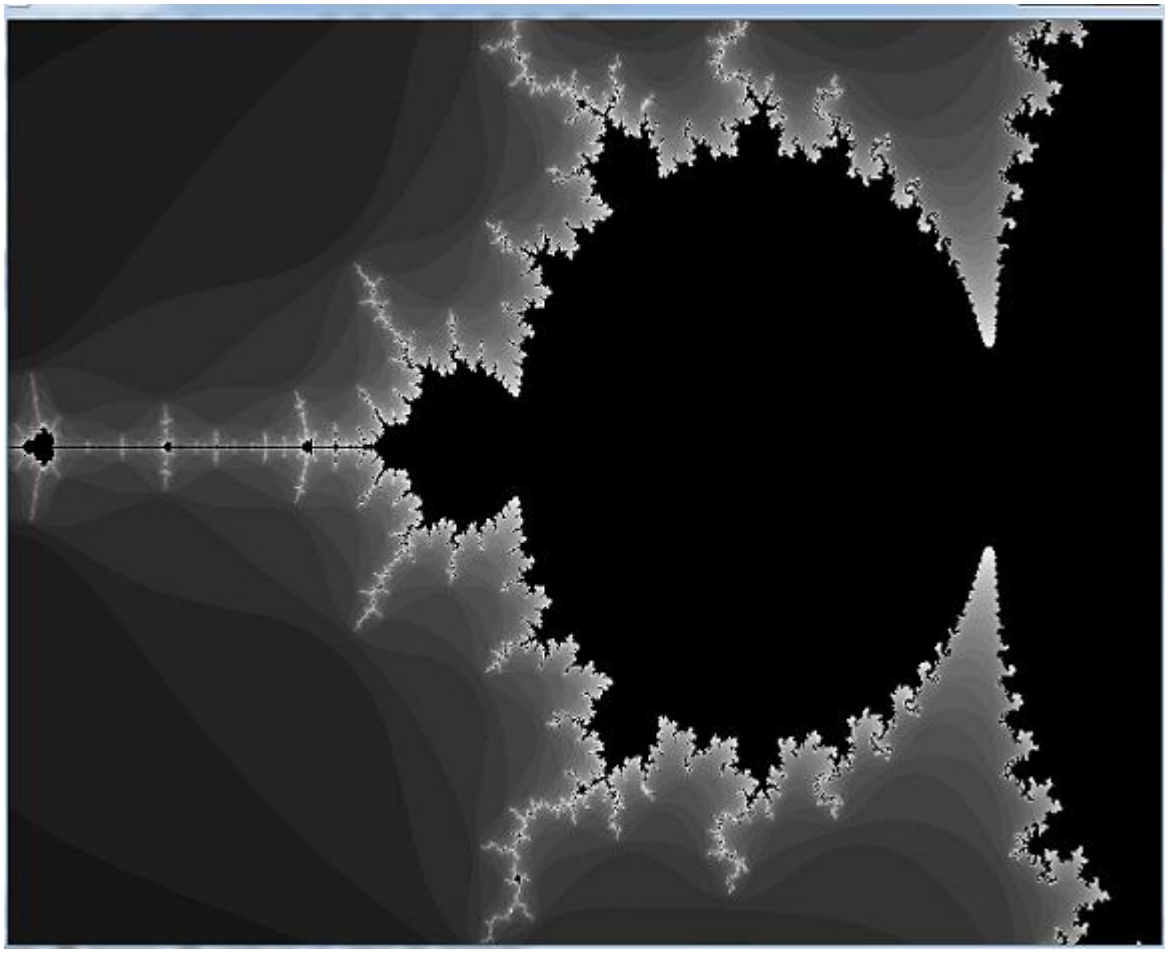

Figure 4 - Mandelbrot after zooming

Mandelbulb

Note that iteration = 32

With nOrder = 2

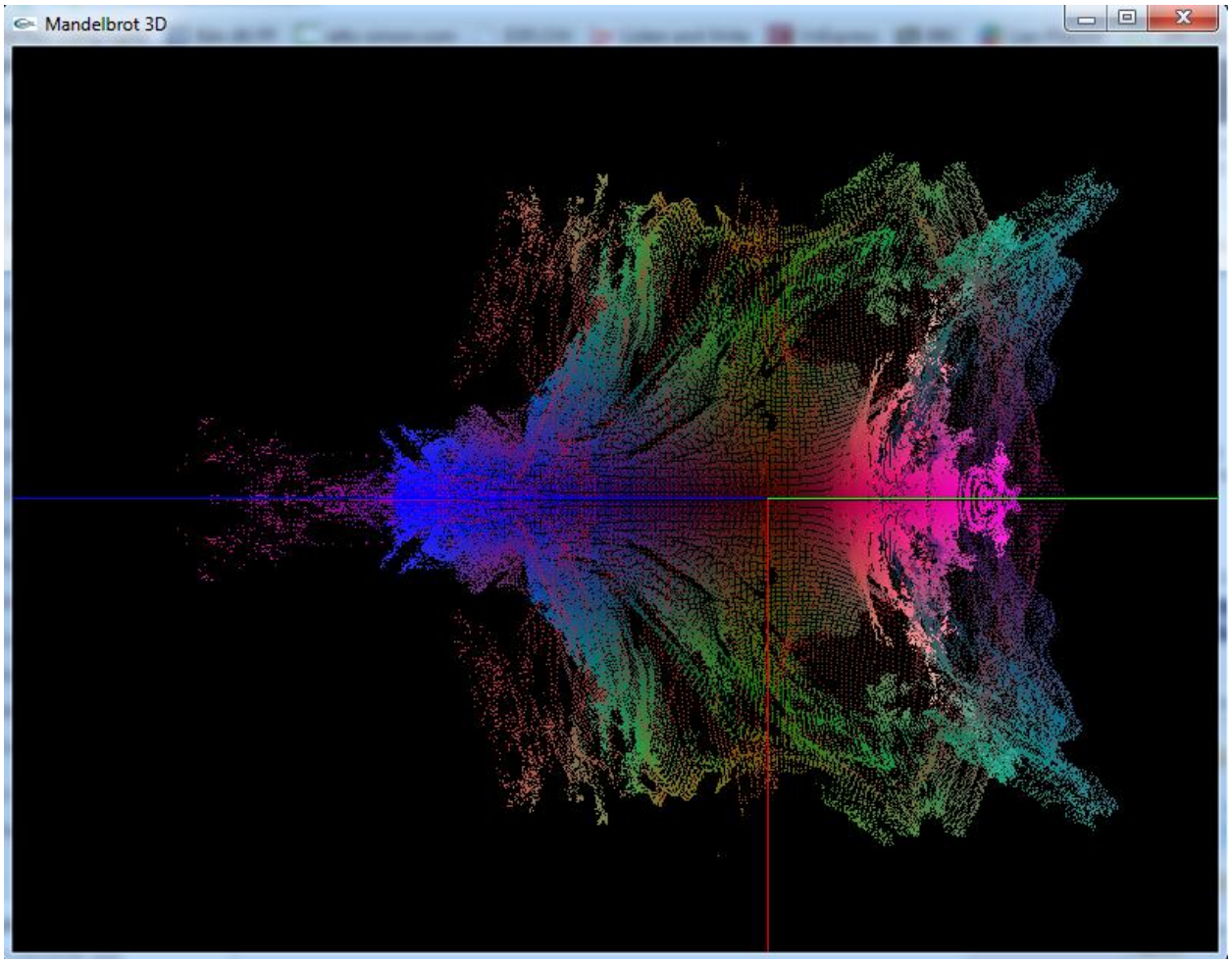

Figure 5 - Mandelbulb with nOrder = 2

With nOrder = 8

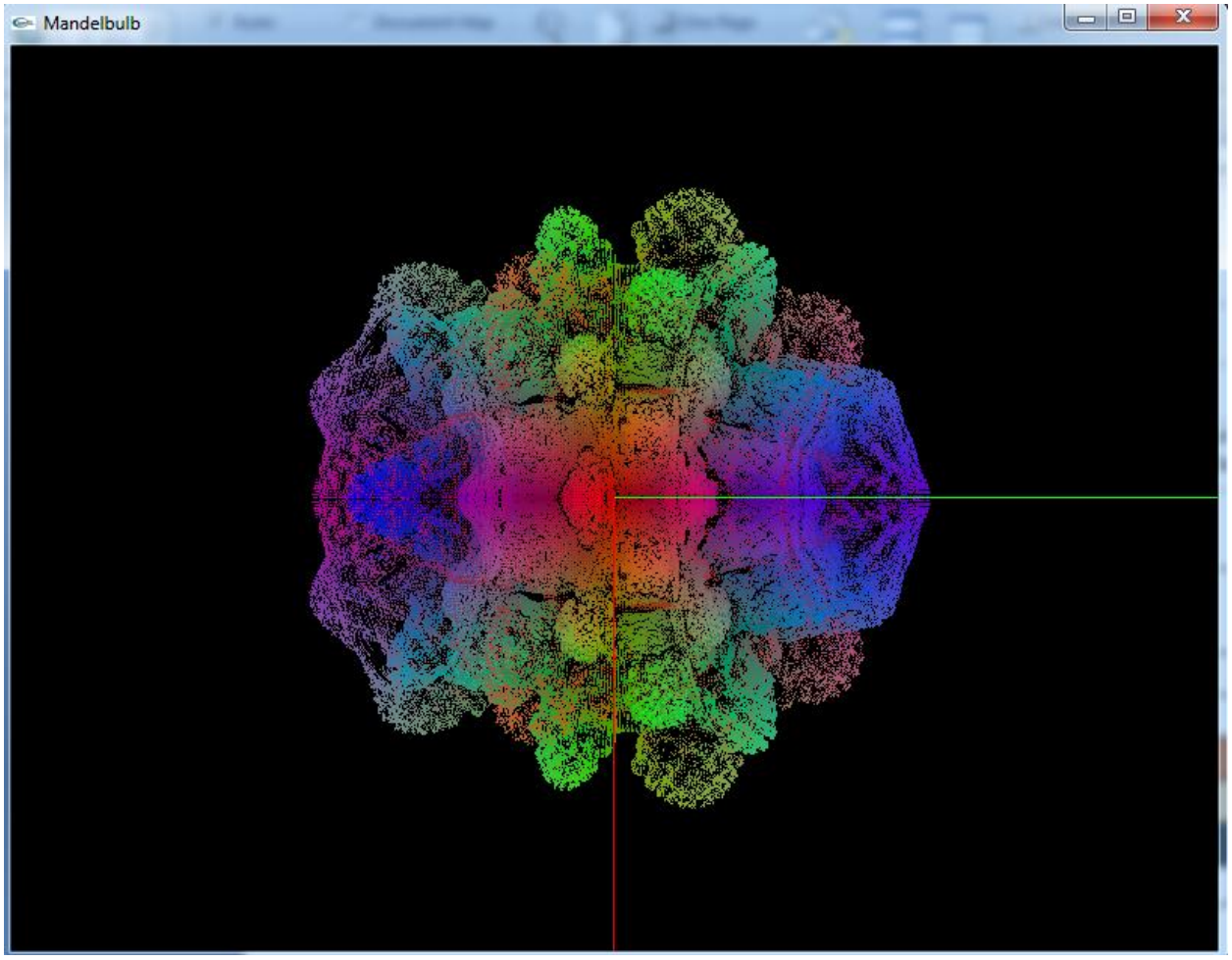

Figure 6 - Mandelbulb with nOrder = 8

Project Plan

Period

About 7 weeks from March 31 to May 21.

Working Plan

Task	Tran DucMuoi	Nguyen ThanhToan	Phan Dang Thanh
Mathematics Overall	Distance Estimator Algorithm, Mandelbulb Formulas	Escape Time Algorithm, Mandelbrot Formulas	Coloring algorithms in both 2 parts
Mandelbrot (2D)		Points Generation	Coloring Points
Mandelbulb (3D)	Points Generation, Camera Views		Coloring Points
Documentation	All works		

Schedule Task

Task	Week					
	1	2	3	4	5	6
Research on Mandelbrot and Mandelbulb set						
Mathematics and formula searching						
Mandelbrot (2D)						
Mandelbulb (3D)						
Coloring Mandelbrot						
Coloring Mandelbulb						
Camera Views on Mandelbulb						
Documentation						

References

1. http://en.wikipedia.org/wiki/Mandelbrot_set : Introduction of Mandelbrot set.
2. http://en.wikipedia.org/wiki/Mandelbrot_set#Computer_drawings : Time escape algorithms.
3. http://fraktal.republika.pl/cpp_sdl.html : Mandelbrot Coloring Algorithms.
4. http://en.wikipedia.org/wiki/Mandelbulb : Introduction of Mandelbulb.
5. http://www.subblue.com/blog/2009/12/13/mandelbulb : Mandelbulb Formulas.
6. http://en.wikipedia.org/wiki/Cartesian_coordinate_system : Cartesian 3D Coordinates.
7. http://www.fractalforums.com/theory/summary-of-3d-mandelbrot-set-formulas/ : Mandelbulb Formulas References.
8. http://www.fractalforums.com/theory/ : Theory about Mandelbulb set.
9. http://www.skytopia.com/project/fractal/mandelbulb.html : Mandelbulb Summary.
10. http://blog.hvidtfeldts.net : More in distance estimator of 3D Mandelbrot set.
11. http://www.fractalforums.com/programming/basic-3d-fractal-coloring/ : Mandelbulb Coloring Algorithms.