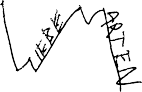
Bachelor project Computing Science – Starting form

*To be filled in and signed by the student and the supervisor(s) at the beginning of the bachelor project.*

Course code: WBCS13000

Credits (EC): 15 points (420 hours)

1. **General information**



Student name and signature: Wiebe-Marten Wijnja

Student number: s2776278

Project title: Efficiently rendering Iterated Function Systems when zooming in

Date: 2020-05-11

Starting meeting: 2020-05-08

Final assessment: expected on 2020-08-03  
  
First supervisor: Jiri Kosinka   
  
Second supervisor: Gerben Hettinga (? waiting for decision)

1. **Project description:**

Iterated Function Systems (IFSs) are a method of constructing fractals [Zob92]. The resulting fractals are often self-similar. Until now, very little research has been done in efficiently rendering IFS-generated fractals when altering the camera angle that looks at such a fractal, such as when zooming in. The contractive and self-similar nature of IFSs might provide some interesting opportunities to improve their rendering process.

In this project, the student will investigate this and come up with new ways to speed up the rendering of IFS-generated fractals, in the hopes of creating an algorithm that runs fast enough on modern consumer hardware to render IFS fractals in high detail in real-time.

The main research question of this project is: “*Is it possible to render animations in which a camera zooms in on an iterated function system fractal in real-time?*”

The most commonly used algorithm to do IFS fractal rendering is the ‘chaos game’ [Bar88]. This is a non-deterministic algorithm that converges to a deterministic result because of the contractive nature of IFSs. Because of its simplicity, it can run very fast. However, the chaos game is usually used in a way in which one draws immediately pixels on a canvas, which means that no calculations can be re-used between to-be-rendered animation frames.

Another disadvantage of the chaos game is that it is not trivial to use in current GPU (OpenGL) pipelines. Some prior work that uses GPGPU techniques to run the chaos game algorithm on GPUs exists [Gre05]. While showing promising efficiency (rendering 1 million points at 20 frames per second), this technique still leaves room for further improvement because it requires (re-)evaluating every IFS-function for every point for each frame.

**References:**

[Zob92] Zobrist, George Winston, and Chaman Sabharwal, eds. *Progress in Computer Graphics.* Vol. 1. Intellect Books, 1992. (p. 135-141)

[Bar88] Barnsley, Michael F. *Fractals everywhere*. Academic press, 2014.

[Gre05] Green, Simon G. "GPU-accelerated iterated function systems." *ACM SIGGRAPH 2005 Sketches*. 2005. 15-es.

1. **Methodology and timeline:**

To create an algorithm that works efficiently with graphics systems, a language environment should be used that:

* compiles directly to machine code, which will make it easier to benchmark rendering speeds;
* is able to communicate well with a GPU, which will be of paramount importance in speeding up the rendering process.

Based on these requirements, the project is expected to be implemented using a general-purpose programming language like Haskell, Rust and/or C++, and using OpenCL or CUDA to program the GPU.

The student has the necessary hardware (a modern Graphics card) available.

**Timeline:**

|  |  |  |
| --- | --- | --- |
| **Week** | **Work** | **Milestones** |
| 1 | Detailed literature study | Come up with ways to improve IFS rendering |
| 2 |
| 3 | Design and prototype | Create a prototype to validate these new rendering ideas |
| 4 |
| 5 | Refining the prototype, improving efficiency  Prepare thesis outline | Improve the prototype to make it more efficient and easier to use/understand  Outline submitted to supervisor |
| 6 |
| 7 |
| 8 |
| 9 | Writing thesis draft | Draft submitted |
| 10 |
| 11 | Writing final thesis, presentation slides | Final thesis and code submitted; project presentation |

The student will be embedded in the Scientific Visualisation and Computer Graphics group of the Bernoulli Institute, and attend selected online meetings of the group.

1. **Division of tasks**

This project will be done by a single student.

1. **Deliverables:**

* BSc thesis;
* Software source code;
* Source code documentation;
* Example images/animations of generated IFS-fractals;
* Final project presentation.

1. **Grading**

Scientific quality of Research and technical contribution: 40%

Project management and interpersonal skills: 20%

Final Presentation: 20%

Report/Thesis: 20%