

## ABV-Visualisierung-Projekte (Stand 2015-09-04)

Die Projektvorschläge werden am **7.9.2015** morgens in der Vorlesung vorgestellt. Die Projekte sollen in 2er Gruppen bearbeitet werden. Bitte stimmen Sie sich mit Ihrem Projektpartner ab und schicken Sie gemeinsam bis **13 Uhr am 8.9.2015** zwei Projektwünsche (1. + 2.) an Konstantin Poelke ([konstantin.poelke@fu-berlin.de](mailto:konstantin.poelke@fu-berlin.de)). Am späteren Nachmittag werden die Projektverteilungen ausgegeben.

## Durchführung eines Projektes

- **Projektauswahl:** Bis um 13 Uhr am Dienstag, 8. September, zwei Projektwünsche (1 Hauptwunsch + 1 Alternative) an Konstantin Poelke ([konstantin.poelke@fu-berlin.de](mailto:konstantin.poelke@fu-berlin.de)).
- **Präsentation:** Am Donnerstag, 17. September, werden im Nachmittagsbereich, ab 14 Uhr, die Projekte präsentiert. Jede Projektpräsentation soll einen Überblick über das Problem (10 Min), die Lösung und Resultate mit Bildern, etc. (10 Min), sowie 10 Min Zeit für Fragen bieten. Die Präsentation soll an Hand von ca. 10 Folien (insbesondere nicht anhand der Projektdokumentation) erfolgen.
- **Dokumentation:** Die schriftliche Dokumentation/Ausarbeitung soll das theoretische Problem und dessen praktische Lösung beschreiben, sowie den Quellcode bzw. die Benutzung des Programms. Die Dokumentation sollte einen Umfang von 8-10 Seiten haben und als PDF vorliegen. Abgabe bis 25. September an Konstantin Poelke ([konstantin.poelke@fu-berlin.de](mailto:konstantin.poelke@fu-berlin.de)).
- **Bewertung:** Die Präsentation (ca. 10 inhaltliche Folien) und die Dokumentation (ca. 8-10 Seiten PDF) gehen je zu 50% in die Bewertung ein.

## Software/Hardware

Sämtliche nötige Software wird am Montagnachmittag, 07. September, eingerichtet. JavaView kann hier heruntergeladen werden: <http://javaview.de/download/index.html>. Maya kann in einer studentischen Lizenz gratis genutzt werden (<http://www.autodesk.com/education/free-software/maya>), oder auf einem der FU-Rechner, an denen Maya installiert ist. Die Geomagic Software liegt auf einem der FU Rechner.

## Projects

### 01 – Lorenz Attractor Camera Ride



**Abstract:** The Lorenz Attractor was derived from a simplified model of convection in the earth's atmosphere. It also arises naturally in models of lasers and dynamos. The movement of a particle is described by a system of non-linear differential equations. The behavior of the particle is chaotic and its path depends on the initial position.

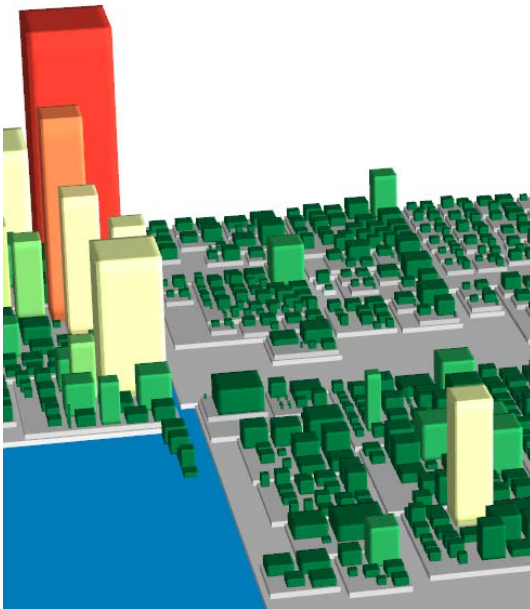
**Software, Literature:**

- JavaView or Maya
- Peitgen: "Chaos and Fractals"

**Expectation:**

- Calculate and visualize the Lorenz attractor as a curve in 3D.
- Attach a camera to the curve and visualize a ride along the curve.

### 02 – Tree Maps



**Abstract:** Tree maps display hierarchical (tree-structured) data as a set of nested rectangles. Each branch of the tree is given a rectangle, which is then tiled with smaller rectangles representing sub-branches. A leaf node's rectangle has an area proportional to a specified dimension of the data. Often the leaf nodes are colored to show a separate dimension of the data.

**Software, Literature:**

- JavaView
- Shneiderman, "Tree visualization with Tree-maps: A 2d space-filling approach"

**Expectation:**

- Implement Tree maps for Java code in JavaView
- Advanced: Implement McCape and other complexity metrics

### 03 – Splatting-Shader for point cloud rendering



**Abstract:** Modern OpenGL is shader based and there are several shader stages responsible for different things that need to be done in the rendering pipeline (rendering polygons, filling polygons with colors, mapping textures, computing new geometries, normals, etc.). This project is to implement a splatting shader for point clouds, which renders each point as a (bend) disk aligned with a normal given at the point.

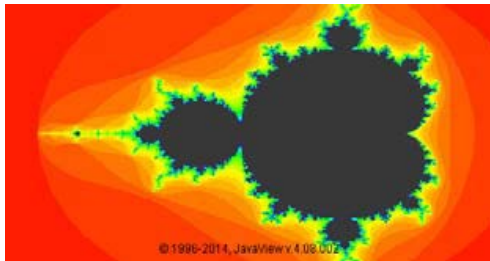
**Software, Literature:**

- JavaView, JOGL, OpenGL 4.3

**Expectation:**

- Implement Splatting shader in the JavaView JOGL Project
- This is an advanced project for people with modern OpenGL knowledge.

### 04 – Mandelbrot



**Abstract:** 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.

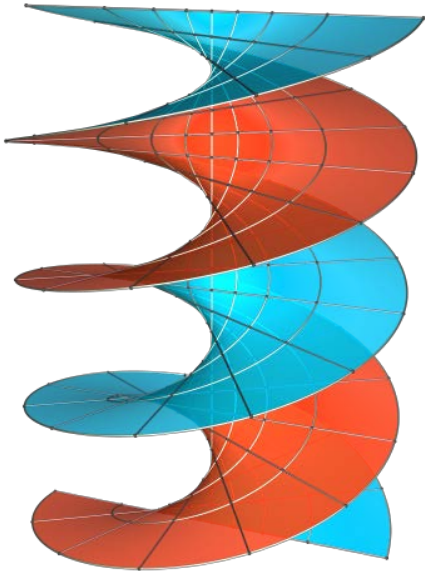
**Software, Literature:**

- JavaView or Python
- Mandelbrot Applet

**Expectation:**

- Implement the Mandelbrot fractal with various color schemes in JavaView or Python

## 05 – Maya Modelling and Animation



**Abstract:** Maya is a 3D computer graphics software used for 3D-animation, modelling, simulation and rendering. It contains a range of sophisticated tools for creating characters and effects, allowing higher productivity in modelling, texturing and creating shaders. This project is to learn modelling and animation with Maya. Therefore generate a model of the helicoid, create a nice visualization of it, and animate a slide of an object of your choice along the helicoid.

### Software, Literature:

- Maya
- Optional: JavaView to generate the model

### Expectation:

- Calculate and visualize a helicoid
- Animate a slide of an object along the helicoid
- Optional: Render the animation.

## 06 – Scanning and Printing



**Abstract:** Scan a model with our 3D scanner and perform some post-processing on the output data. This includes merging several partial scans together using the GeoMagic software. The point cloud should then be thinned out, discarding unnecessary points.

### Software, Literature:

- Geomagic and corresponding handbook

### Expectation:

- Scan a shape and perform postprocessing on the point cloud
- Advanced: print the model in 3D