Web Based Terrain Modeller

MSc in Computational and Software Techniques in Engineering

Digital Signal and Image Processing

Miguel Marques

Supervisor: Dr. Peter Sherar

22nd July 2016

www.cranfield.ac.uk

Outline

1. Problem
2. Relevant Background
3. Methodology
4. Software
5. Demonstration
6. Benchmarks
7. Conclusion & Further Work

Problem

Functional Requirements

- Procedurally generate detailed terrains
- Use a CAD surface as a base

Technical Requirements

- Web-based
- Based on fractal geometry

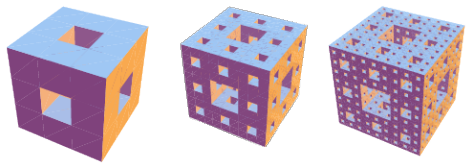
Additional Requirements

- Real-time editing

Relevant Background

Fractals

in Weisstein, Eric W. "Koch Snowflake." From MathWorld – A Wolfram Web Resource. http://mathworld.wolfram.com/KochSnowflake.html



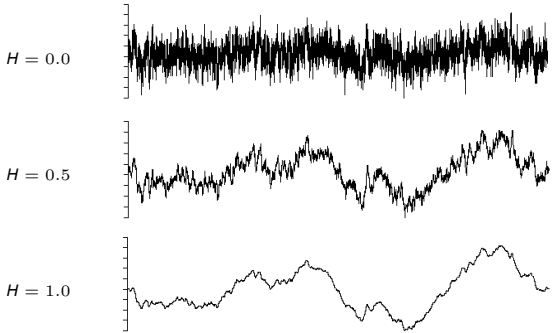
in Weisstein, Eric W. "Menger Sponge." From MathWorld – A Wolfram Web Resource. http://mathworld.wolfram.com/MengerSponge.html

Relevant Background

Fractional Brownian Motion (fBm)

$$D_f = D_E + 1 - H$$

D_f Fractal dimension
D_E Euclidean dimension
H Hurst Exponent



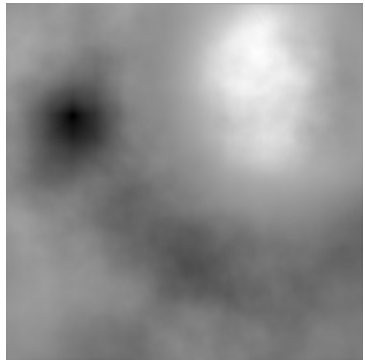
in http://paulbourke.net/fractals/noise/

Relevant Background

Terrain Representation

Height Map

- Two dimensional array
- Each position saves an altitude value
- Can be saved as a grayscale image



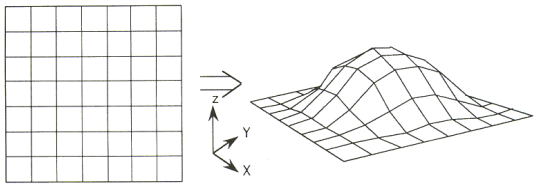
Example of height map

Relevant Background

Terrain Representation

Regular Grid

- Wireframe processing
- Rendering



in http://wtlab.iis.u-tokyo.ac.jp/ wataru/lecture/rsgis/rsnote/cp6/cp6-10.htm

Relevant Background

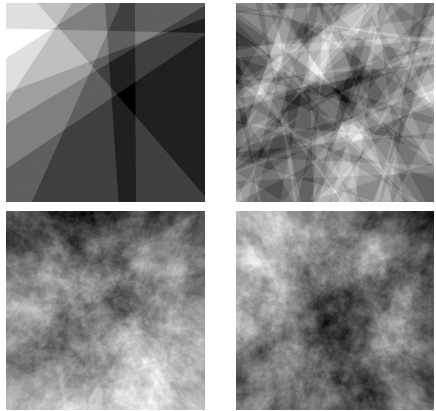
Terrain Generation Methods

- Poisson Faulting
- Subdivision Methods
- Fourier Filtering
- Noise Synthesis

Relevant Background

Terrain Generation Methods

- **Poisson Faulting**
- Subdivision Methods
- Fourier Filtering
- Noise Synthesis

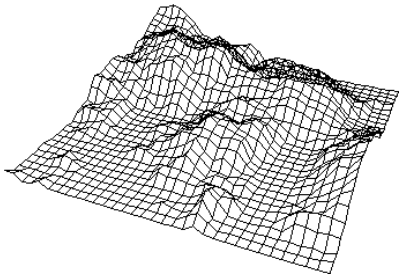


in http://paulbourke.net/fractals/noise/

Relevant Background

Terrain Generation Methods

- Poisson Faulting
- **Subdivision Methods**
- Fourier Filtering
- Noise Synthesis



in http://paulbourke.net/fractals/noise/

Relevant Background

Terrain Generation Methods

- Poisson Faulting
- Subdivision Methods
- **Fourier Filtering**
- Noise Synthesis

$$\beta = 1 + 2H \Leftrightarrow H = \frac{\beta - 1}{2}$$

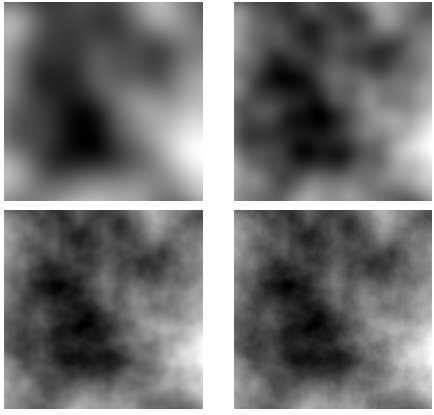
$$D_f = D_E + 1 - H = D_E + \frac{3 - \beta}{2}$$

D_f	Fractal dimension
D_E	Euclidean dimension
H	Hurst Exponent
β	Filter Power

Relevant Background

Terrain Generation Methods

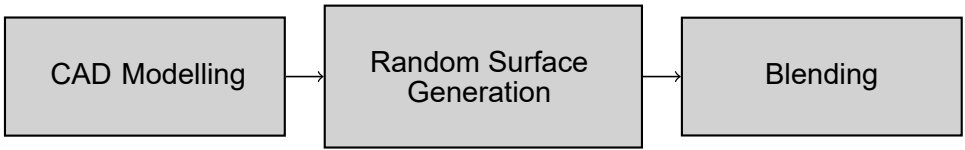
- Poisson Faulting
- Subdivision Methods
- Fourier Filtering
- **Noise Synthesis**



Perlin Noise Synthesis with 1, 2, 4 and 8 octaves

Methodology

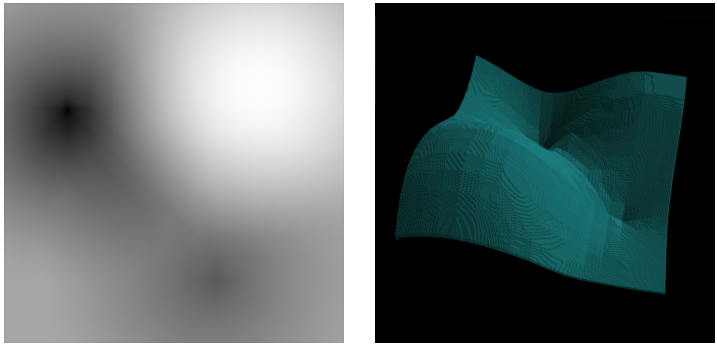
Overview



Process Phases

Methodology

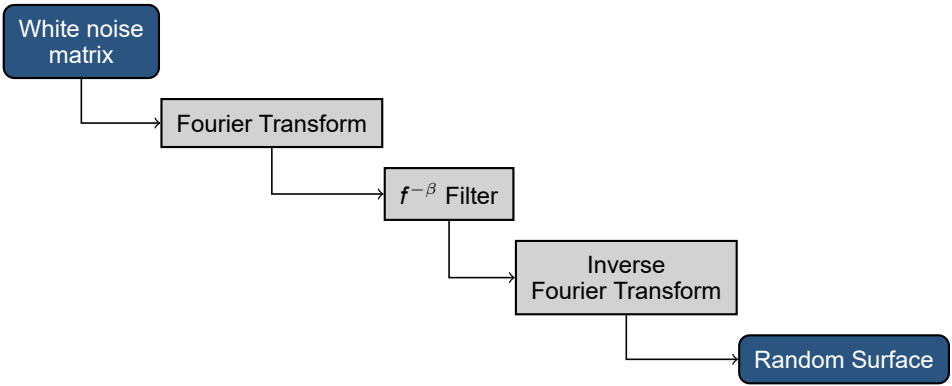
CAD Modelling



Example output from CAD Modelling Phase

Methodology

Random Surface Generation - Fourier Filtering



Methodology

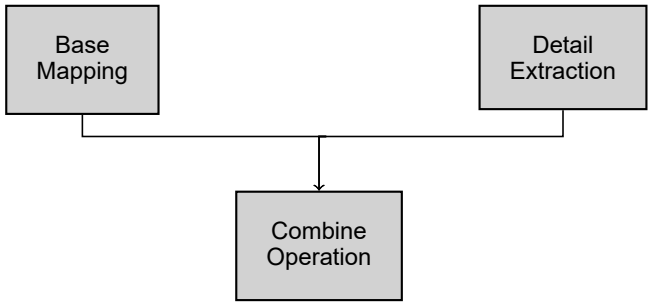
Random Surface Generation - Noise Synthesis

$$\frac{\sum_{i=0}^{O-1} noise(x \times L^i + B, y \times L^i + B) \times P^i}{\sum_{i=0}^{O-1} P^i}$$

O	octaves	Number of frequencies
L	lacunarity	Gap between successive frequencies
P	persistence	Contribution gap between successive octaves
B	base	Frequency displacement
	noise	Noise function (eg. Perlin or Simplex)

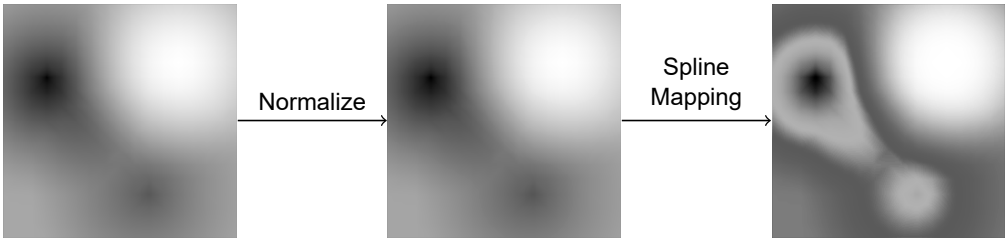
Methodology

Blending Process



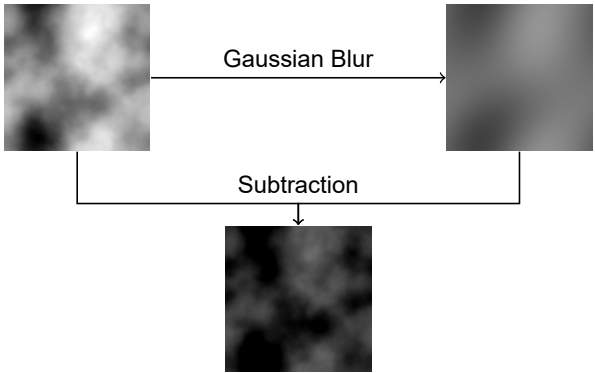
Methodology

Blending Process - Base Mapping



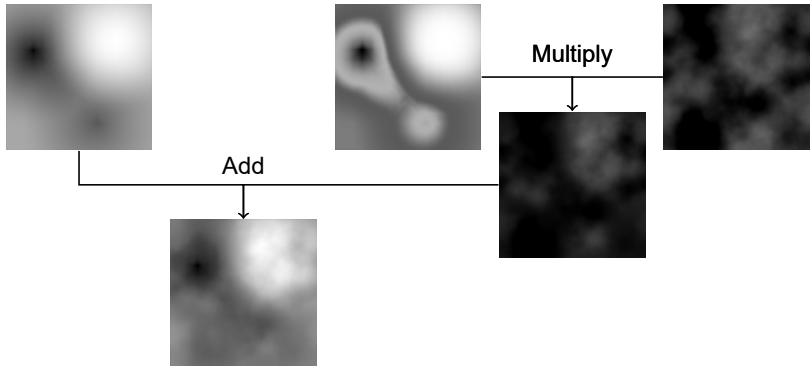
Methodology

Blending Process - Detail Extraction



Methodology

Blending Process - Combine Operation



Software

Technologies

three.js

Software

GPU Computations

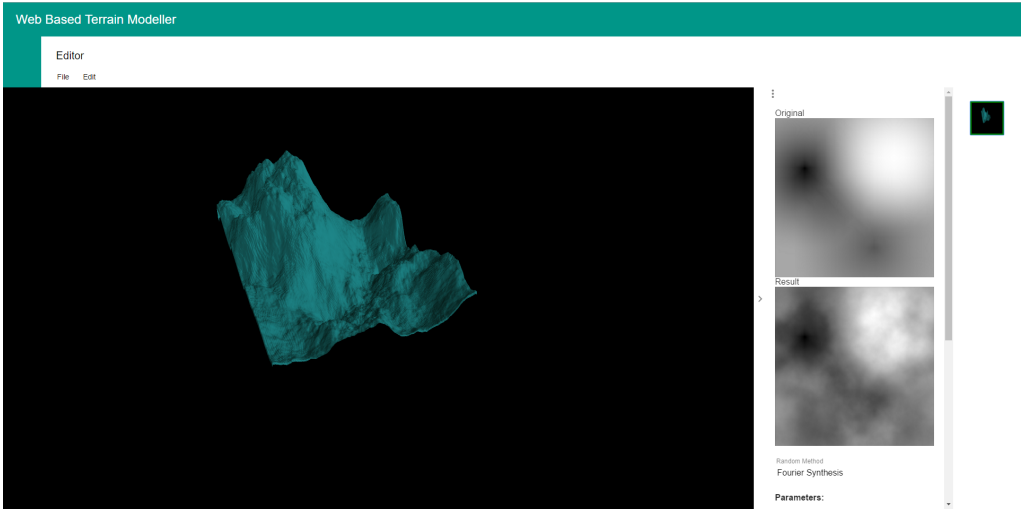
Technical Details

- WebGL 2
- Render to Texture
- Floating-point textures

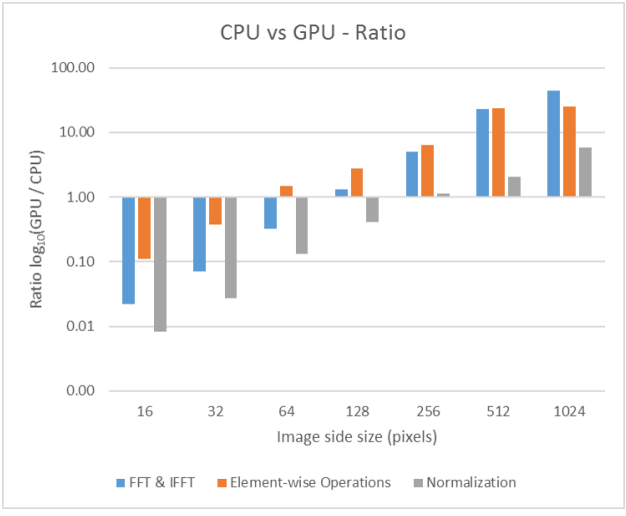
Features

- FFT and IFFT
- Element-wise operations
- Matrix Normalization

Demonstration



Benchmarks


CPU vs GPU - Ratio
100.00
10.00
1.00
0.10
0.01
0.00
Ratio log10(GPU / CPU)
16
32
64
128
256
512
1024
Image side size (pixels)
FFT & IFFT
Element-wise Operations
Normalization

Conclusion & Further Work

What has been done

- Hybrid Process for terrain modelling
- Web-based implementation
- WebGL 2 used for computations

Future Work

- Different detail extraction methods
- Real world data