Seminar 2016
J. Schmidt and E. Gr´oller
(Organizers)

Volume 1 (2016), Number 1
STAR – State of The Art Report

State-of-the-Art-Report for the Seminar

Bernhard Rainer[1] and Silvester Farda[1]

[1]TU Wien, Austria

Figure 1: *Report teaser*

Abstract
// TODO WRITE ABSTRACT

Categories and Subject Descriptors (according to ACM CCS): Computer graphics [Computing methodologies]: Rendering—

1. Introduction

Terrain modeling and representation is a elaborated topic in computer graphics and visualization. Traditional content creation requires a vast part of development recources. Therefore it is only plausible to look for new solutions, that allows for faster, more accurate and physically plausible terrain generation and representation. This report elaborates a handful of different solutions, that approach seeral diffent tasks in terrain generation. We will highlight several techniques on terrain modeling, datastructures and rendering approaches. Furthermore this report presents several solution for more physically correct erosion simulations based on hydrology and heat transfer, as well as tectonic plate movement

2. Terrain Modeling from Feature Primitives

In [GGP*15] a practical approach is described to model terrains using a construction tree, whos leave nodes are feature primitives. Such primitives describe landmark features, like rivers, mountains, valleys, lakes, and more universal features, like valleys, roughness, hills. The construction tree combines these primitives using a set of operators, which describe the type of combination.

2.1. Primitives

A primitive can either be image based, or skeletal-based. Both primitives feature an elevation function and a weight function that describe the height of the terrain, as well as the potential interaction with other primitives. Skeletal-based primitives profit from a short render time and smaller memory footprint, whereas image-based primitives havea higher memory cost, but provide an easier way to integrate real data into the scene.

2.1.1. Skeletal primitives

Skeletal primitives are defined by a geometric sceleton(point, segment, curve or contour) and a set of parameters that describe the elevation and the weight function.

A **disc primitive** contains several parameters, such as the center c and radius r, describing the area of influence, as well as a noise function $\eta(p)$, which controls the local ground roughness. $\{a_i\}$ is a set of decreasing amplitudes, $\{s_i\}$ a set of increasing frequencies. Combining these parameter one can obtain the elevation function

$$f(p) = c_z + \sum_{i=0}^{n-1} (a_i \eta((p-c)s_i)).$$

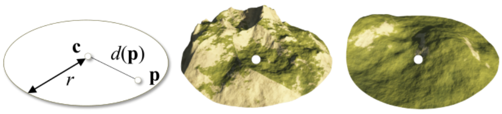

Figure 2: *This figure shows a disc primitive creating a circular landform. The mountain is obtained using ridge-multifractal noise function, smooth hills are created using a turbulence function.*

Curve primitives are made up of a piecewise curve skelekton Γ and a set of profiles $\{c_i\}$. A profile describes a cross section perpendicular to the sceleton. Any function can be used to describe the cross section, in this example each cross section described as a one-dimensional quadratic function. The position is then constructed by interpolation using the curvilinear coordinates.

One large use of curve primitives is rivers. A river follows its path, described with the curve skeleton. Using different profiles a homogenous shore can be obtained. More complex profiles can describe more larger river systems like deltas and side streams. Modifiying the profile accordingly, one can simulate roads with this model, by parametrising the cross section with the road width and defining the elevation of the road in order to allow combination with other primitves in the terrain.

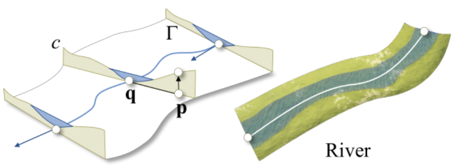

Figure 3: *A curve primitive is used to create a river. q corresponds to the projection of p onto the skeleton.*

Contour primitives describes a closed curve around a center point and a set of profile curves $\{c_i\}$. Much like with curve primitives the profile curves discribe the elevation in radial direction. This approach allows to compose more complex features.

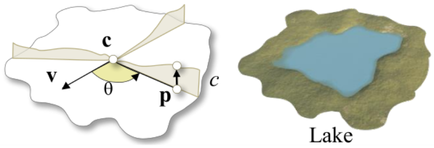

Figure 4: *A contour polygon describes a complex system of polygons. In this figure it is used to describe a lake.*

2.1.2. Image primitives

Complex terrain features are difficult to create procedurally. To add complex feature, such as detailed river shores or sand ripples, image based primitives are used. This process is similar to using a heightfield on a terrain.

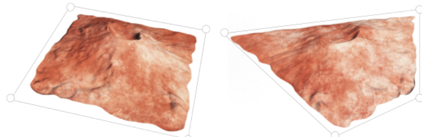

Figure 5: *Real data is mapped onto a quadrangle that can be deformed arbitrarily.*

2.2. Operators

Much like in Constructive Solid Geometry the inner nodes of the construction tree are operators. These operators combine the elevation function f and weight function α of their sub-trees. For simplicity we consider binary nodes and denote the two sub-trees A and B.

2.2.1. Blending

The elevation of two nodes are mixed according to their corresponding weight function. This allows to combine large scale terrain primitives in an effective matter. If two primitives are far away, the do not influence each other. The resulting terrain is the union of to sub-trees. If their regions of interest intersect they blend together.

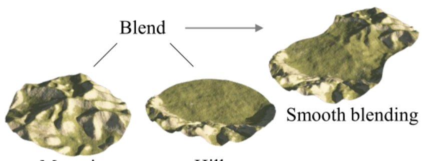

Figure 6: *This figure shows a blending of a mountain and hill to a smoother landscape. the areas of interest intersect partly.*

2.2.2. Replacement

This operator defines specific local changes in the terrain. Such changes for example are lakes, rivers, roads. If the areas of interest intersect, the elevation in A is replaced with the value in B.

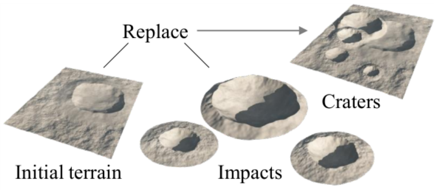

Figure 7: *Replacement is used to construct a lunar landscape..*

2.2.3. Addition

The addition operator is used to add variations and details to the terrain. The additional elevation to f_A is controlled by the product of f_B and α_B

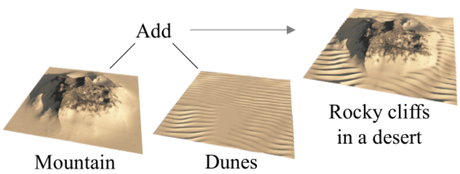

Figure 8: *Real data is mapped onto a quadrangle that can be deformed arbitrarily.*

2.2.4. Warping

The warping operator allows to distore the shapeoif a surface by displacing the elevation and weight with a certain value, obtained by the warping function. Unlike the other operators, this operator only works on one sub-tree. Applying a warp operator to a sub-tree can mask visual artifacts due to repetitive usage of primitives.

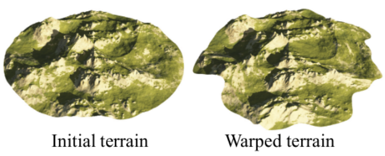

Figure 9: *This figure shows a deformed terrain using the warping operator.*

2.3. Rendering

The paper propeses two different methods of rendering the tree structure. An advanced approach on Sphere Marching allows for high quality images at an interactive framerate. The second algorithm uses adaptive quad tree tesselation with dynamic level of detail. The algorithm allows for multi resolution and crack-free terrain generation. Furthermore techniques like view frustum culling and distance based adaptive tesselation can be used to reduce computational costs.

3. Fast Hydraulic Erosion Simulation and Visualisation on GPU

The behaviour of water massively influences the look and features of a terrain. Depending on the timespawn, amount of water, ground composition and many more factors, hydraulic erosion creates a variety of different, but distinctive ground deformations, such as ripples, ridges, meanders, riverruns and valleys. This paper [NWD05] presents an approach capable of creating such deformations in short amount of time, making it suitable for interactive visualization. The model represents terrain and water surfaces using heightfield, which is sampled onto a regular grid. A shallow water model is used to update the water surfaces and the fluid velocity field.

3.1. Water simulation

This paper uses a simplified Navier-Stokes equation to simulate the water flow. The approach discards 3D-features like mutiple water layers, vertical vortices and waves. this makes the equation less physically correct, but much faster to compute. The appraoch is based on a system of first order differential equations. This system describes the movement of material, in this case the amount of water at each cell, depending on velocity $\vec{v}$ and acceleration $\vec{a}$. Since this approach is dealing with complex erosion, the system uses am multidimensional material vector storing additional parameters like dissolved sediment.

$$\dot{\vec{v}} = \vec{a} - K_A \cdot \vec{v} = \frac{\vec{F}}{m} - K_A \cdot \vec{v}$$

$$K_A$$

describes the friction between the fluid and the terrain and can be manipulated for test purposes.

A vector field assumes that the particles movement is constant. In a common Newtonian Physic System objects are accerlated by the gravity. The direction of acceleration is the direction of the biggest tile angle α of the underlying height field. Therefore the acceleration force can be computed from the sinus of α times the gravitational constant g. The angle α is determined by the gradient $\nabla I(x,y)$ of the height field.

At this point he acceleration direction is only defined in x and y direction. The acceleration vector $\vec{M}$ can be computed using the gradient $\nabla I(x,y)$:

$$\vec{M} = (-\frac{\Delta I}{\Delta x}, -\frac{\Delta I}{\Delta y}, -\frac{\Delta I^2}{\Delta x^2}, -\frac{\Delta I^2}{\Delta y^2})^T$$

The acceleration vector can now be computed as follow:

$$\vec{a} = \frac{|\vec{M}_z|}{|\vec{M}|} \cdot g \cdot \frac{\vec{M}}{|\vec{M}|}$$

$\vec{M}_z$ describes the acceleration direction direction along the z axis.

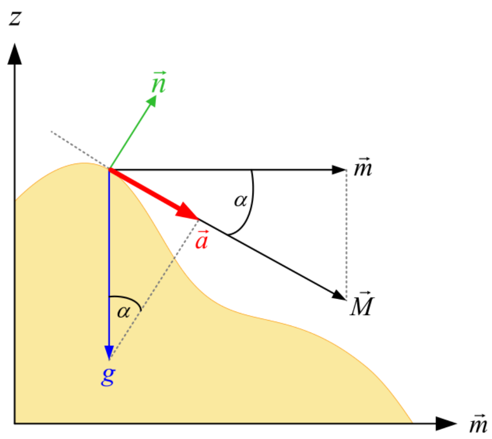

Figure 10: *Calculation of acceleration.*

3.2. Hydraulic erosion function

When water flows, soil, stones and other materials are dislocated and transported to lower regions. To simulate this effect the paper defines a hydraulic erosion function. Whenever a part of a material is moved to another grid cell by the simulation, this function is used. It determines the amount of disolved and deposited parts of material.

References

[GGP*15] GÃĽNEVAUX J.-D., GALIN E., PEYTAVIE A., GUÃĽRIN E., BRIQUET C., GROSBELLET F., BENES B.: Terrain modelling from feature primitives. *Computer Graphics Forum 34*, 6 (2015), 198–210. URL: http://dx.doi.org/10.1111/cgf.12530, doi:10.1111/cgf.12530. 1

[NWD05] NEIDHOLD B., WACKER M., DEUSSEN O.: Interactive physically based fluid and erosion simulation. In *Proceedings of the First Eurographics Conference on Natural Phenomena* (Aire-la-Ville, Switzerland, Switzerland, 2005), NPH'05, Eurographics Association, pp. 25–33. URL: http://dx.doi.org/10.2312/NPH/NPH05/025-032, doi:10.2312/NPH/NPH05/025-032. 3