Johns Hopkins
Engineering for Professionals

605.767 Applied Computer Graphics

Brian Russin

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Module 7H
Fractal Terrain and Diamond Square

Fractal Terrain Generation

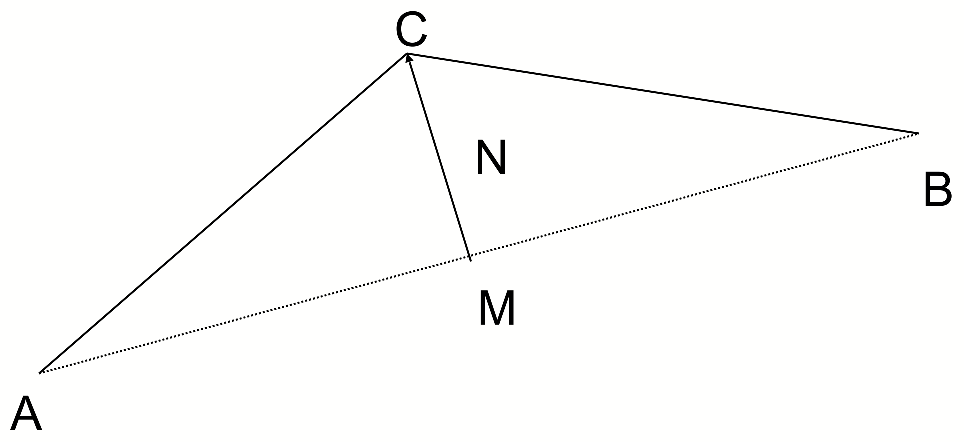
- Fractal terrain generation is a subdivision technique to generate 3D terrain as a polygon mesh object
  - Can generate a large number of polygons
  - Application can control the depth of the subdivision
  - Does not require any special rendering treatment
    - Pass into a standard polygon based rendering engine
- Start with a coarse approximation to the terrain
  - e.g., pyramid to generate a single mountain
  - Could use a coarse terrain elevation data set
- Recursively subdivide each facet
  - To a required level of detail
  - Split the polygon into smaller and smaller parts
  - Spatially perturb these parts
    - Initial shape is retained to an extent depending on the perturbation applied at each subdivision
    - Higher the level (of subdivision) the smaller the perturbation

Recursive Line Subdivision

- Most common subdivision algorithm developed by Fournier, Fussell, and Carpenter (1982)
  - Recursively subdivides a line segment
- Subdivide the line - perturb midpoint in normal direction

$$C = M + t(B - A)^{\perp} * factor$$

  - Random displacement (t)
    - Perturbation automatically decreases as the level increases
      - Perturbation is scaled by the length of the line
  - Two new lines form an isosceles triangle



Controlling the “Jaggedness” of the Fractal Noise

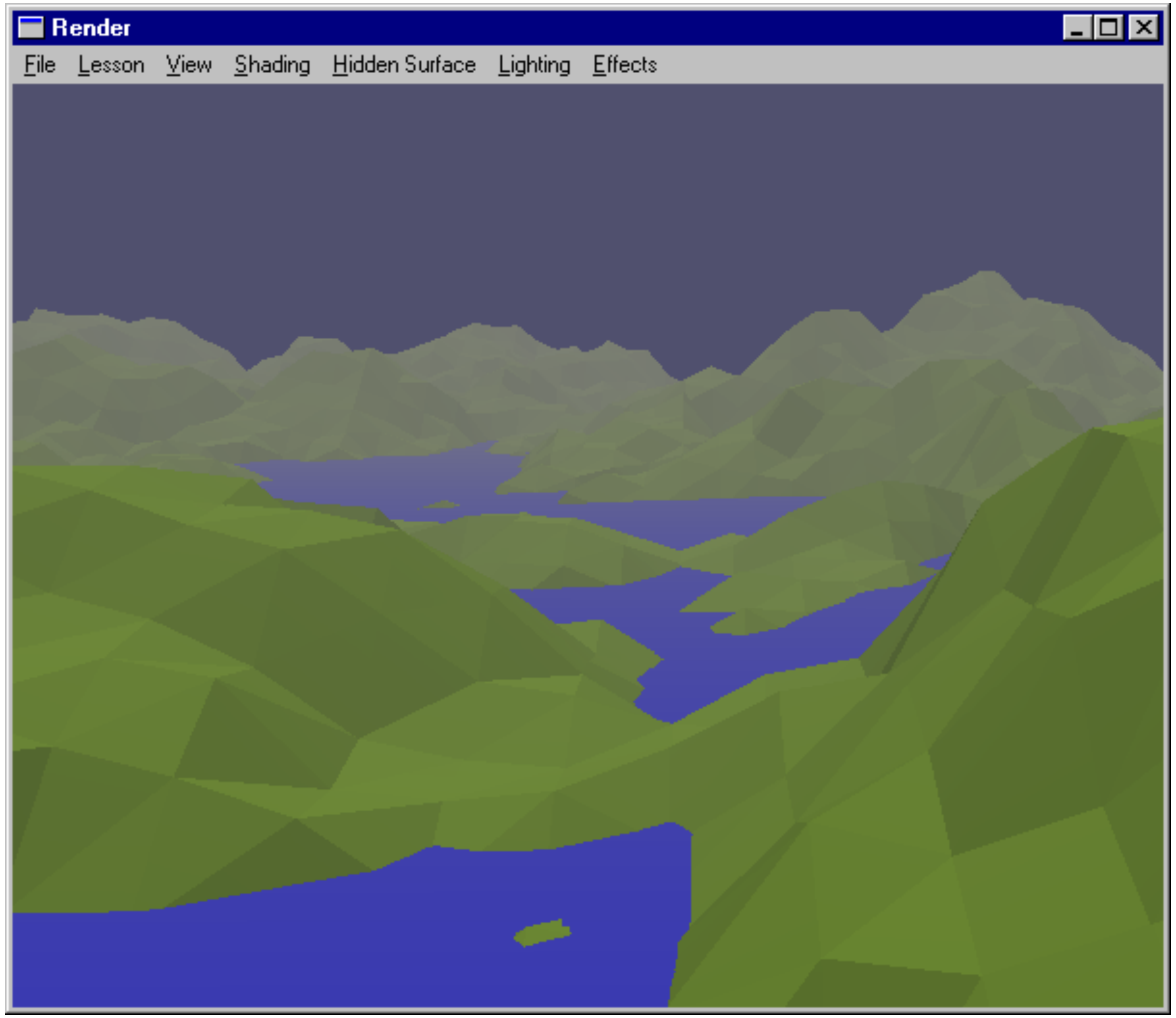
- Use a Gaussian random variable t
- At each level of recursion the standard deviation is scaled by a “roughness” factor
  - Roughness determines extent of the perturbation
  - Fractal curves generated with this method have power spectral density:

$$S(f) = \frac{1}{f^{\beta}}$$

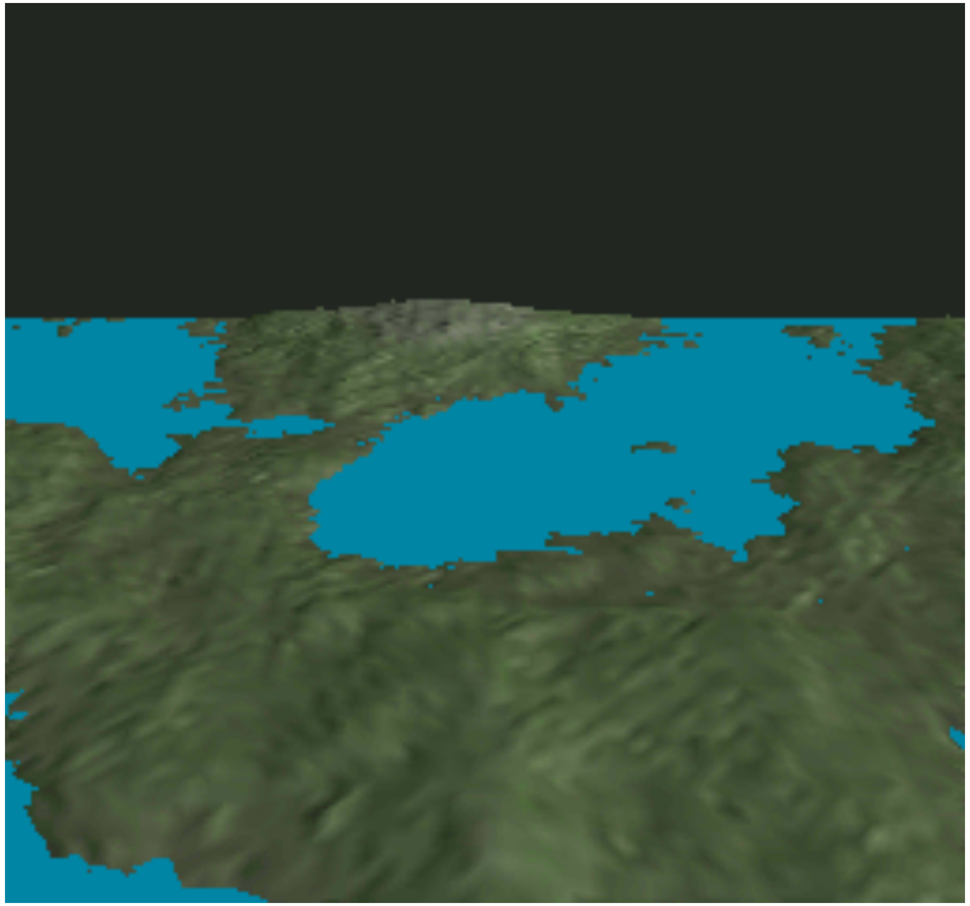
    - β is the power (exponent) of the noise process
  - When β = 2 the process is known as **Brownian motion**
  - When β = 1 the process is called “1/f” noise
  - Fractal dimension D: $D = \frac{5 - \beta}{2}$

Factor	B	D
1.4	1	2
1.1	1.6	1.7
1.0	2	1.5
0.9	2.4	1.3
0.7	3	1

Fractal Terrain With Fog

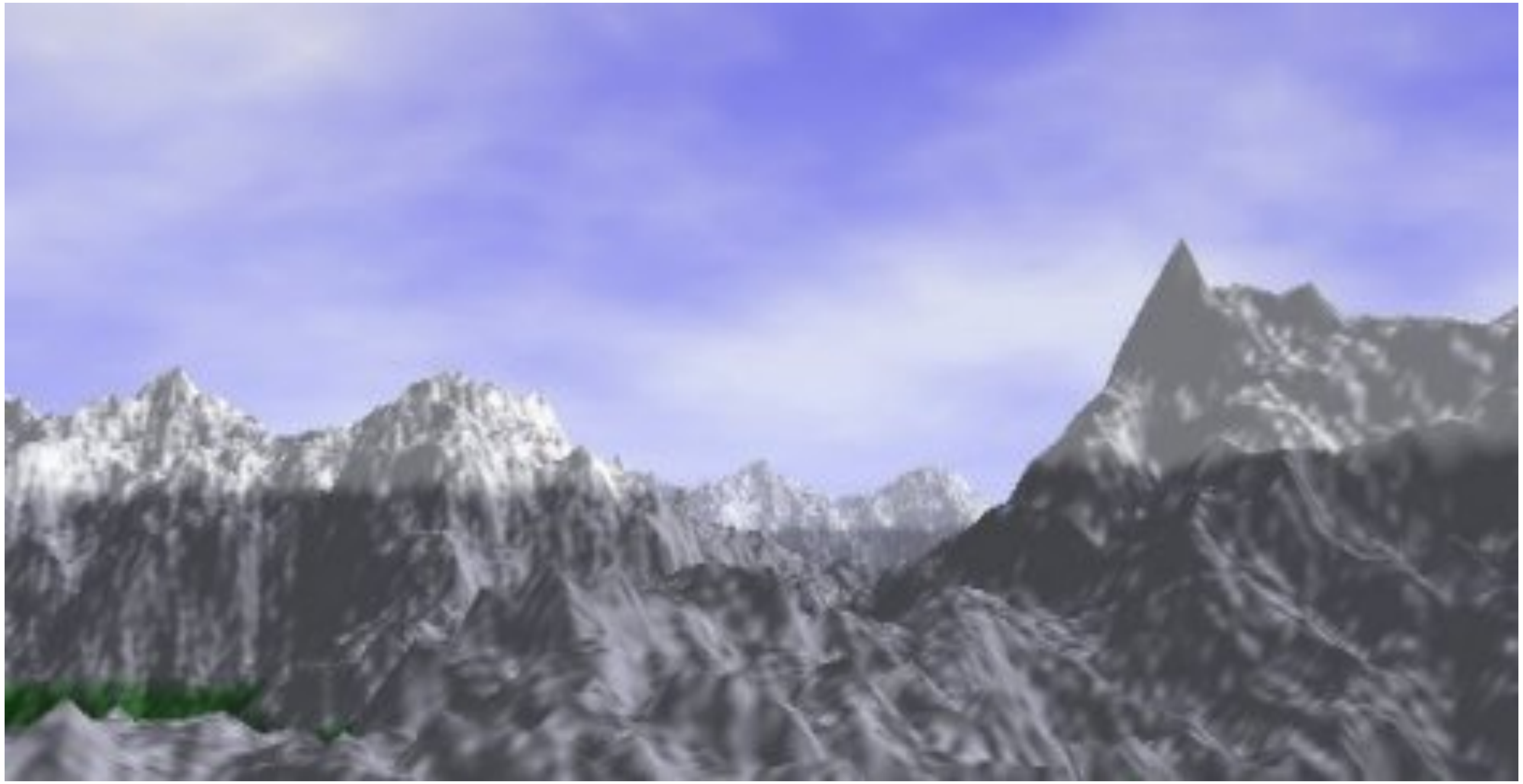


Fractal Terrain



[link no longer active]

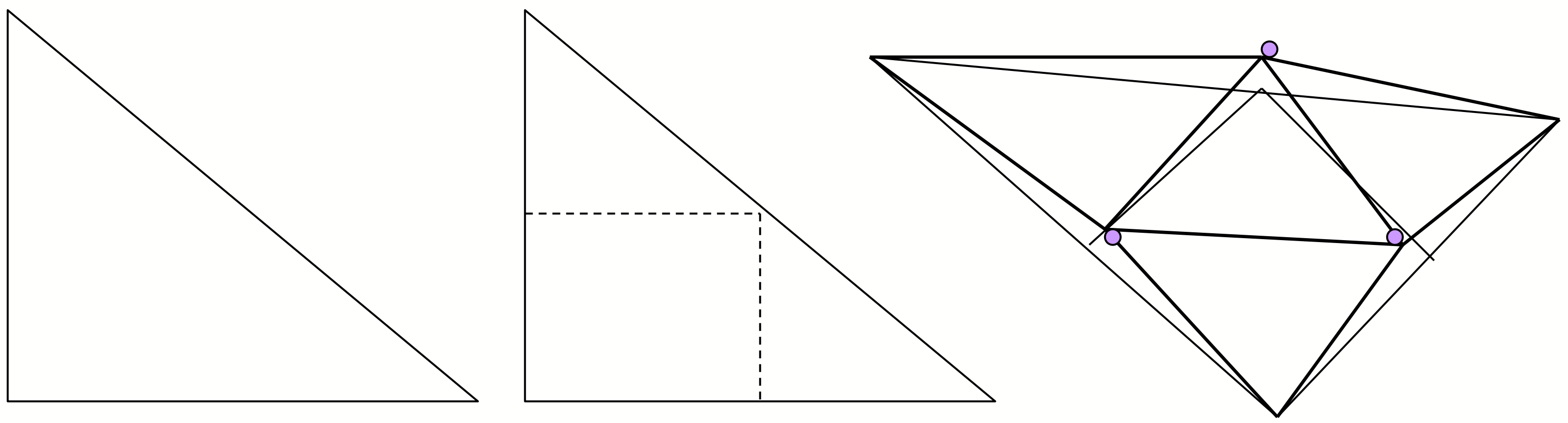
Fractal Terrain



[link no longer active]

Fractal Terrain

- Extend subdivision technique to triangles
  - Triangle represents a terrain facet
- Subdivide the edges
  - Displace each midpoint along a line normal to the original triangle
- Fournier considers two problems
  - Internal consistency
  - External consistency



Internal Consistency

- **Internal consistency** - shape should not change
  - Regardless of orientation which it is generated
  - Coarser details should remain the same if the shape is plotted at higher level of detail
- Gaussian random numbers generated must be unique to the specific point
  - Not a function of the point's position
  - An invariant identifier can be associated with each point
  - Give each point a key value used to index a Gaussian random number generator
  - Hash function used to map the keys of the 2 edge endpoints to a key for the midpoint
  - To preserve scale: the same random numbers must be generated in the same order at a given level of subdivision
  - Noise function

External Consistency

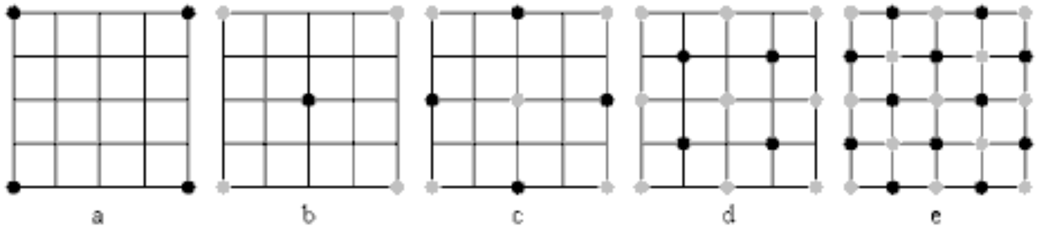
- **External consistency** requires the triangles to be created such that no overlaps or gaps occur
  - Triangles sharing edges must have the same displacement applied
  - Key value and hash function usage creates the same displacement magnitude
- Displacement direction must be consistent for adjacent triangles or gaps or overlaps will occur
  - One solution is to average the normals of the 2 triangles sharing the edge
    - This strategy must be applied at each level of subdivision
    - Can be expensive
  - Cheaper solution - displace all subdivisions along the normal to the original polygon
    - Avoids surface normal calculations at each level of subdivision

Defects in Fractal Terrain

- Creases are a visual defect that can occur in fractal terrain
  - Nature of the model: original polygons tend to be large
    - Creating a detailed model from a coarse original shape
  - Vertex is only displaced once and subdivision edges will align themselves along original polygon edges
    - Creates creases
  - Alignment of edges is noticeable despite the detail
  - Hiding the landscape with fog can decrease the problem
- Animation with fractal terrain can cause problems
  - Polygons can be subdivided to different levels in different frames
    - Results in a bubbling effect
- https://developer.nvidia.com/gpugems/gpugems2/part-i-geometric-complexity/chapter-2-terrain-rendering-using-gpu-based-geometry

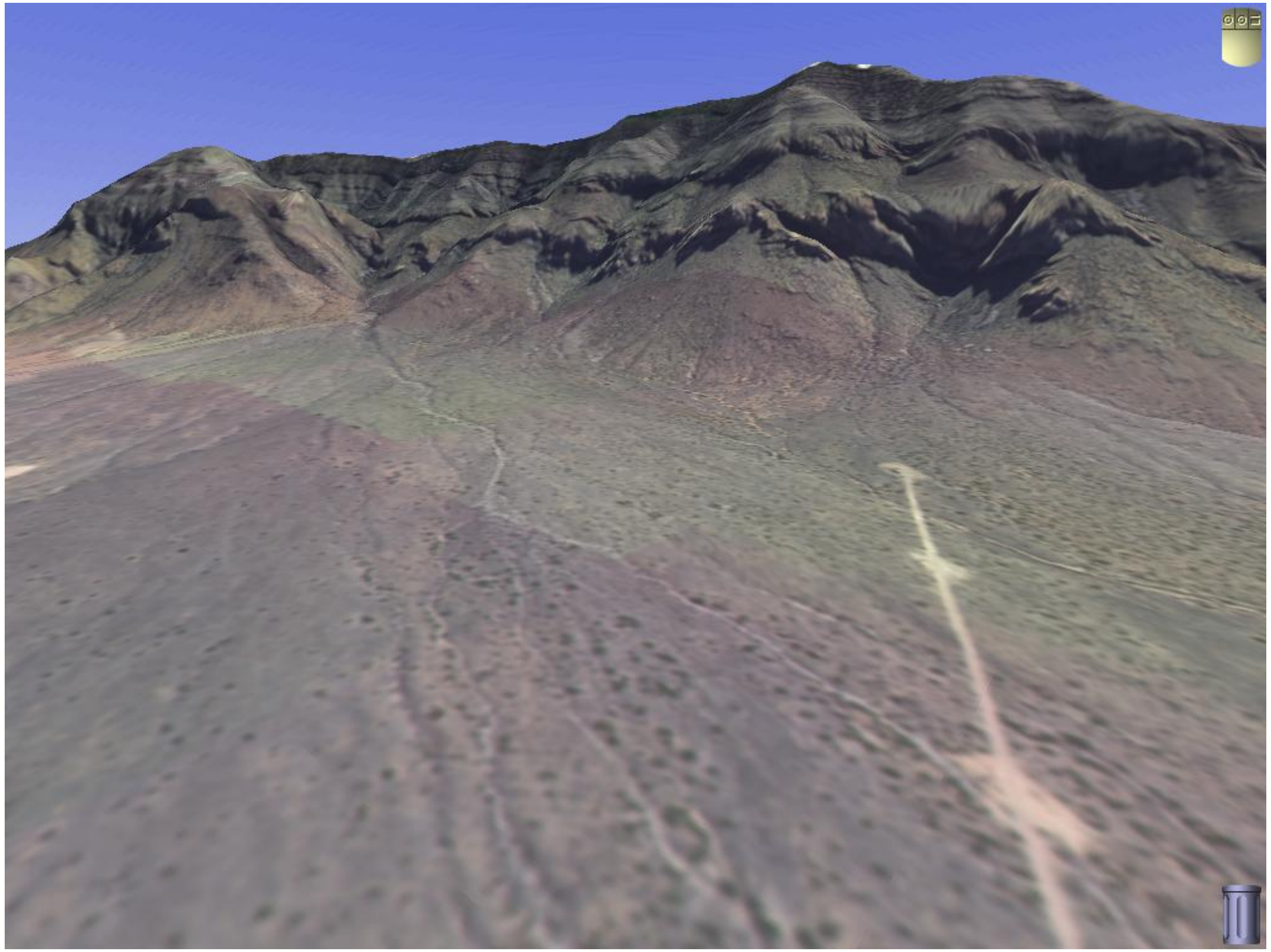
Diamond Square Fractal Terrain Generation

- Use an array that is a power of 2+1
  - Seed the 4 corner points
- Two-step iterative subdivision:
  - Diamond step: generate a random value at the midpoint of the square
    - Average the 4 corner points and add a random value
    - Diamond shapes occur when there are multiple squares (e)
  - Square step: generate a random value at the center of each diamond
    - Average the 4 corner points and add a random value
  - Reduce the range of the random value at each iteration
- Total number of squares is 2^{i+2}, where i = number of iterations



https://computer-graphics.se/TSBK07-files/pdf16/13c%20terrains.pdf

View Dependent Terrain Simplification



http://www.cc.gatech.edu/~lindstro/ (link expired)

View Dependent Terrain Simplification

- Terrain data often has unique properties
  - Typically represented by uniform grid of heights
    - e.g. height posting every 90 meters
  - Often simplified to some level of detail
- Terrain visualization is a difficult problem
  - Lots of research has been performed
  - Many applications require merging accurate terrain with detailed images
    - High frame rates
    - Simulations (flight, military, others), analysis, planning, intelligence systems
- Preferred solutions use view dependent approach
  - Maintain high detail in “foreground”
  - Decrease detail further from eye/camera
    - Especially near “horizon” which can encompass large area
  - Maintain dynamic, view-dependent triangle meshes and texture maps
  - Features such as:
    - View-dependent error metrics and bounds
    - Reduce to specified triangle counts
  - Exploit frame-to-frame coherence

Terrain Software

- http://www.vterrain.org/LOD/Implementations/
- Scape 1.2
  - https://mgarland.org/software/scape.html
  - Algorithms described in the paper Fast Polygonal Approximation of Terrains and Height Fields, by Michael Garland and Paul Heckbert
  - Qslim - simplify arbitrary polygonal models
- Peter Lindstrom and others
  - https://www.cs.princeton.edu/courses/archive/spr01/cs598b/papers/lindstrom96.pdf
  - Real-Time, Continuous Level of Detail Rendering of Height Fields
- ROAM
  - https://www.osti.gov/servlets/purl/632827
  - Real-time Optimally Adapting Meshes
- Many others
  - Numerical Designs Limited: terrain as part of its gaming engine
  - **3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics**
    - Chapter on how to represent terrain