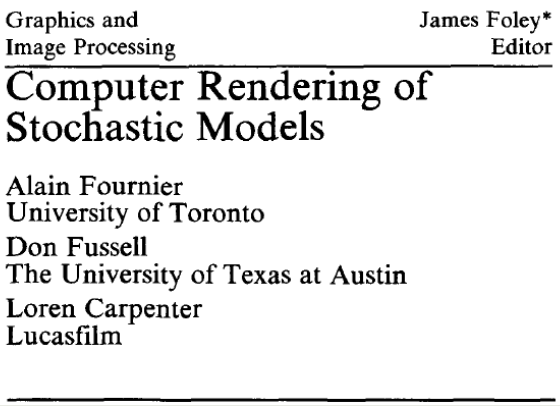
CS 418: Interactive Computer Graphics

Terrain Generation

Eric Shaffer

Terrain Generation

- Lots of scenes require geometric models of natural objects
  - Clouds, Water, Plants, Terrain
- Those last two items are often modeled using fractal techniques
  - Allows for the procedural generation of highly detailed models
- We'll look at a simple fractal modeling technique for terrain
- Diamond-Square Algorithm
  - Developed by Loren Carpenter in 1980(ish)



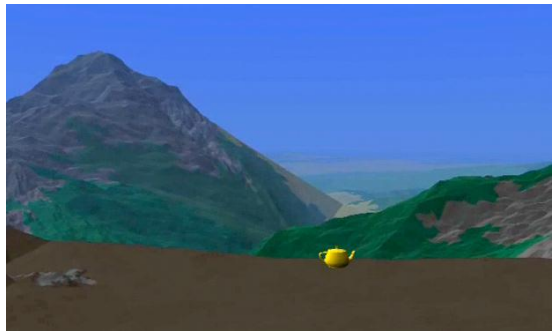
Graphics and Image Processing — James Foley* Editor

Computer Rendering of Stochastic Models

Alain Fournier
University of Toronto
Don Fussell
The University of Texas at Austin
Loren Carpenter
Lucasfilm

Loren Carpenter

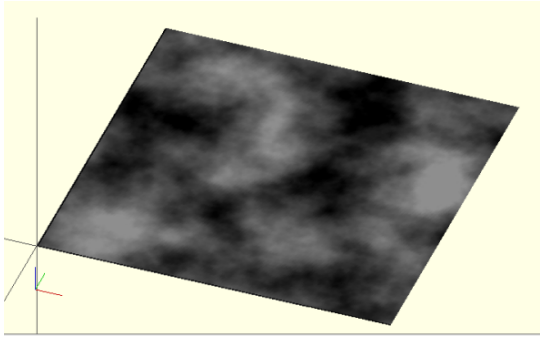
- https://www.youtube.com/watch?v=y5moYMIp8iU
- https://vimeo.com/5810737
- Born in 1947
- Co-founder and chief scientist at Pixar
  - One of the designers of Reyes
  - One of the authors of RenderMan
  - Invented the A-Buffer hidden surface algorithm
  - Improved Mersenne Twistor RNG (2006)
- Retired in 2014



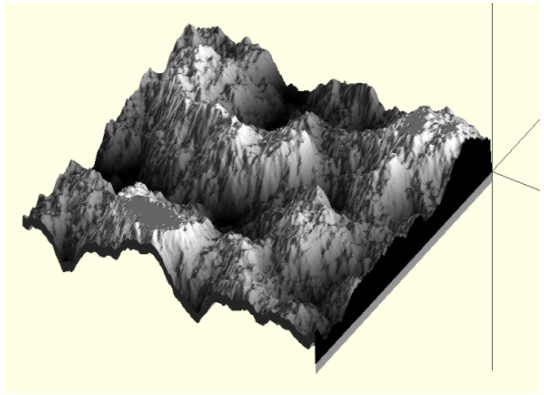
Height Map

- A height map is simply data structure that
  - For each point (X,Y) in a 2D domain
  - Describes a height or Z value

Note: in graphics, Y is usually height
the rest of the world uses Z

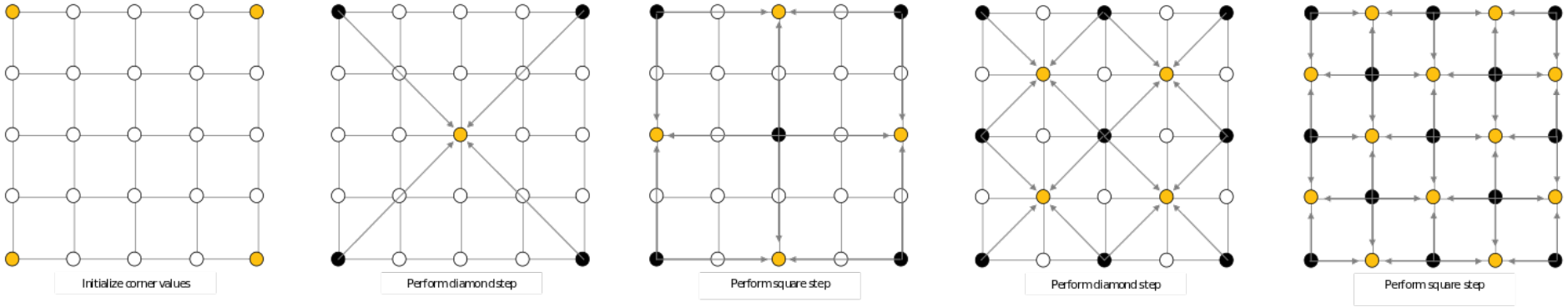


- The (X,Y) points are usually discretely sampled
  - Typically in a uniform grid
  - Images are often used to store height maps



- What phenomenon cannot be modeled this way?

The Diamond-Square Algorithm

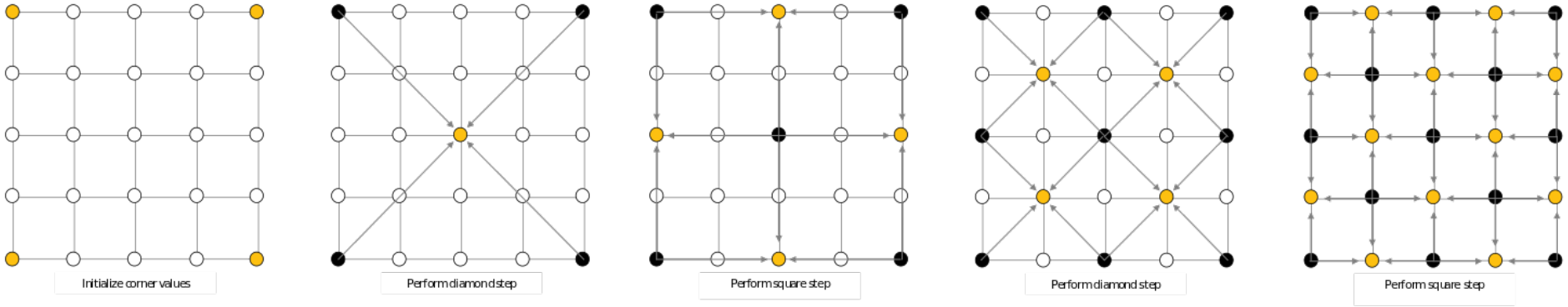


"Diamond Square" by Christopher Ewin - Own work. Licensed under CC BY-SA 4.0 via Commons - https://commons.wikimedia.org/wiki/File:Diamond_Square.svg#/media/File:Diamond_Square.svg

The Diamond-Square Algorithm

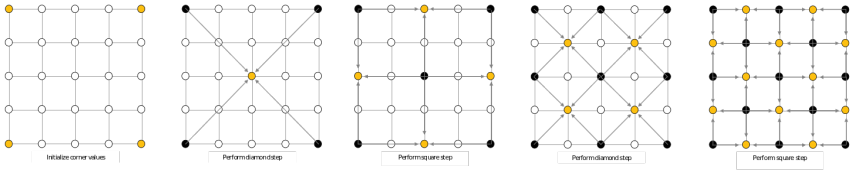
- Create an 2D array of size 2^n+1 by 2^n+1
- Initialize 4 corners to some heights
  - Can choose randomly or hard-code the values
- Until all array values are set:
  - For each square in the array,
    midpoint height = avg four corner points + random value
  - For each diamond in the array
    midpoint height = avg four corner points + random value
  - Reduce the magnitude of the random value
  - Divide each square into 4 sub-squares and iterate

The Diamond-Square Algorithm



"Diamond Square" by Christopher Ewin - Own work. Licensed under CC BY-SA 4.0 via Commons - https://commons.wikimedia.org/wiki/File:Diamond_Square.svg#/media/File:Diamond_Square.svg

The Diamond-Square Algorithm

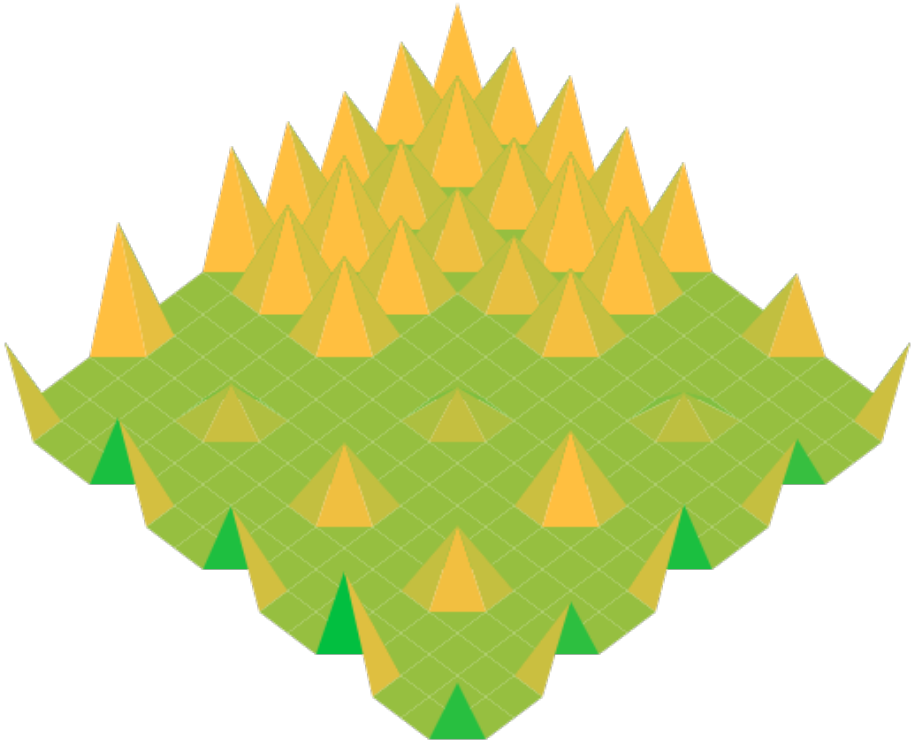


For the square step, the boundary vertices will be set by averaging 3 instead of 4 height values

Alternatively, you can act like the grid is periodic and wrap around to average 4 values

Animation of the Algorithm

http://www.paulboxley.com/blog/2011/03/terrain-generation-mark-one#



Computing Normals

- If you have an indexed face mesh
  - Create an array of normals N, one for each vertex
  - For each triangle (v1 v2 v3)
    - Compute a normal
    - Add the normal to N[v1], N[v2], and N[v3]
  - Normalize each normal N[i]
- If you have a triangle soup…it's more complicated
  - You essentially convert the soup to an indexed mesh
  - Use a hash table to hash 3D points to an array location

Fractal Terrain