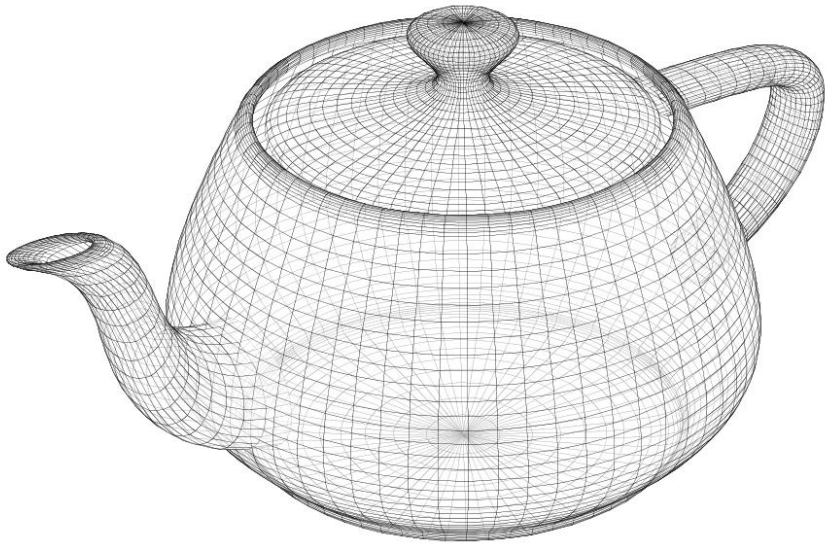


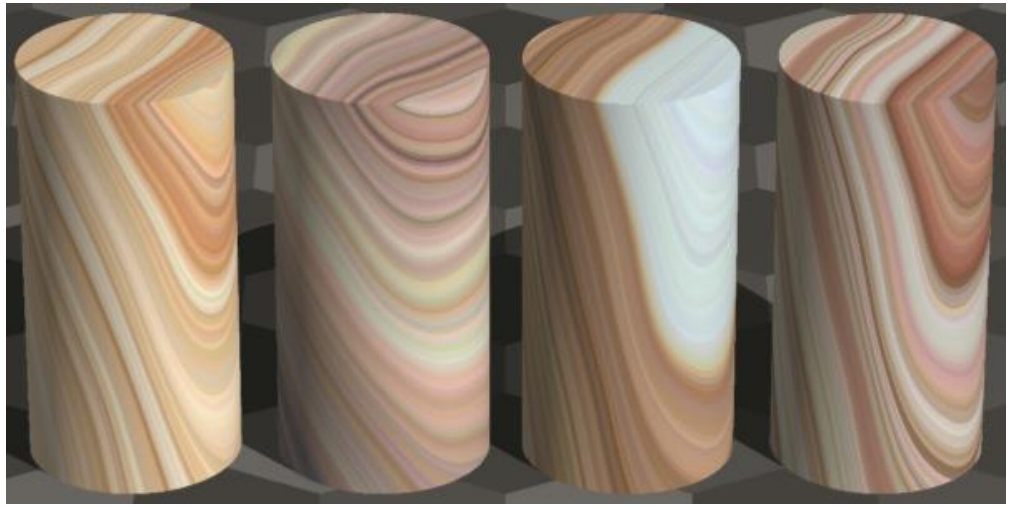
Noise-Based Textures: Noise Functions

Production Computer Graphics

Eric Shaffer

Objectives

- Be able to compute and interpolate noise on a lattice
- Use sums of lattice noises to construct
  - fractal sum, turbulence, and fractional Brownian motion
- Be able to implement marble and sandstone

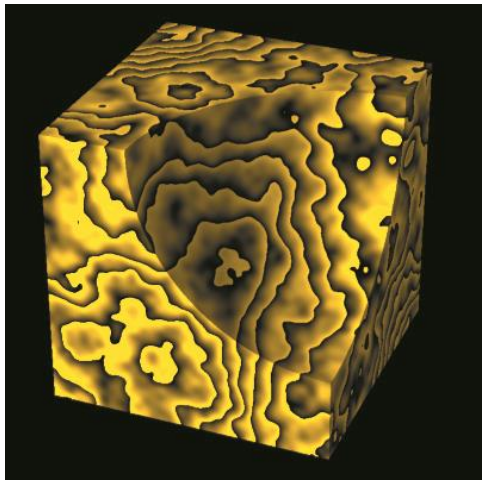


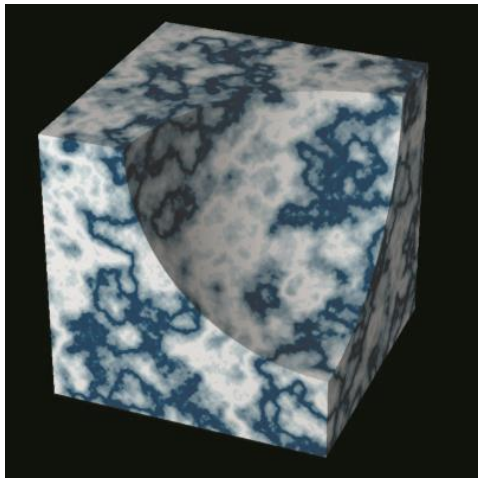
ILLINOIS

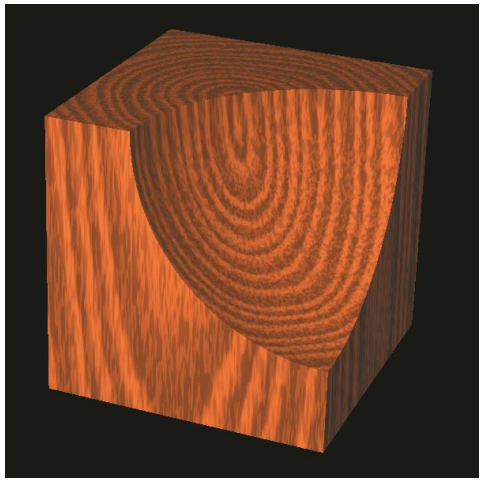
Noise-based Textures

Much like fractal modeling of geometry…

- Noise-based textures model semi-random natural textures







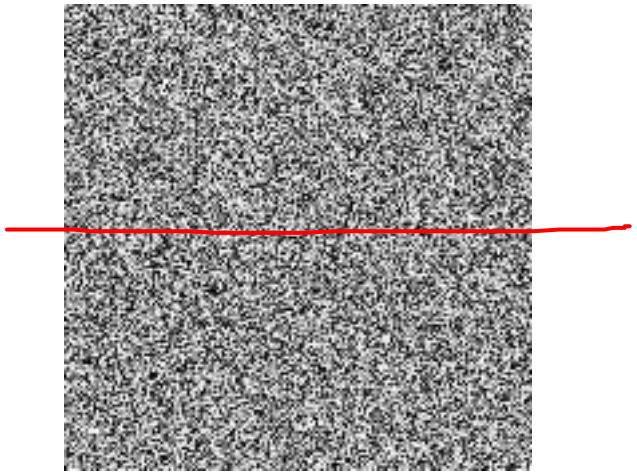
ILLINOIS

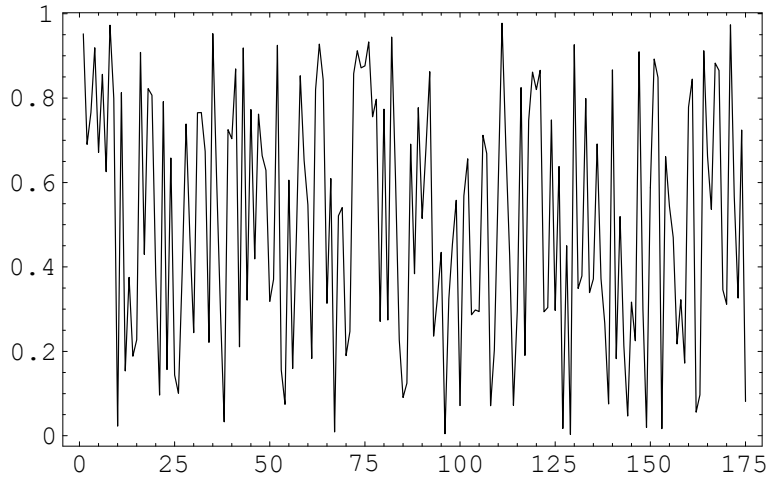
Desirable Properties in Noise

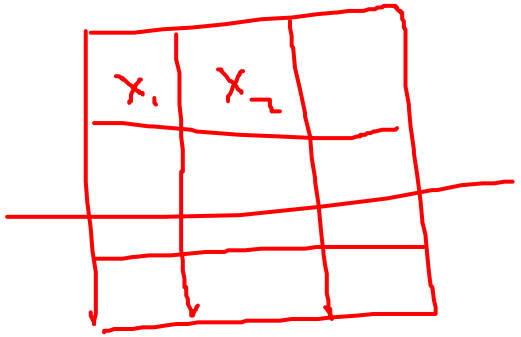
- Generated by repeatable pseudo-random function
- Has a known range
- Band-limited
  - Rate at which the function varies by position is limited
- Not obviously periodic or regular
- Stationary
  - Statistical properties invariant with position
- Isotropic
  - Statistical properties same in all directions

White Noise

- What property doesn't hold?
- Not band-limited…will result in aliasing



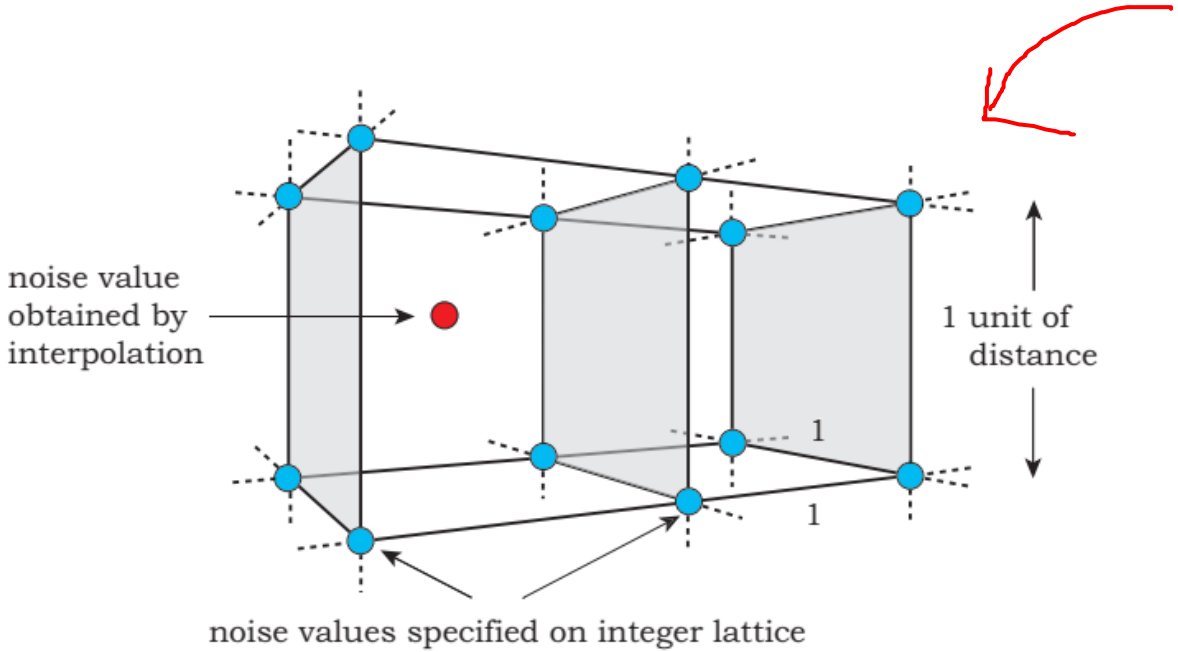


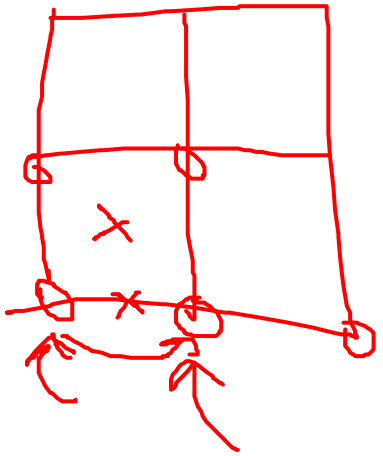


Lattice Noise Functions

Store pseudorandom number at integer coordinate positions

- Use interpolation to generate values not on lattice points





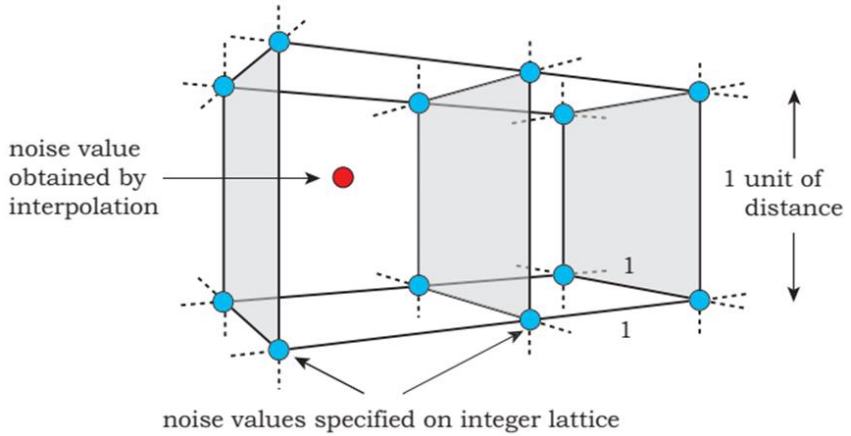
Infinite Lattice

RTftGU uses a 256-value 1-dimensional array of function values

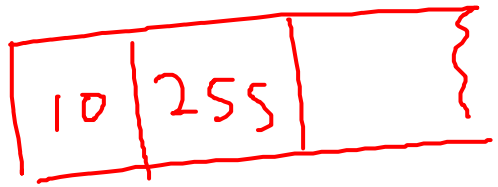
- value_table[i] yields a number in [0.0,1.0]

Access it by mapping (x,y,z) to an integer in [0,255]

Given hit point (x,y,z) find integer coordinates (ix,iy,iz)



- Truncate floating point numbers to integer to get (ix,iy,iz)
- Keep a permutation array PERM with values 0 through 255 randomized
- Hash (ix,iy,iz) into the permutation array using index function:



$$perm(x) = \mathrm{PERM}[x \bmod 256]$$

$$index(ix, iy, iz) = perm((ix) + perm((iy) + perm(iz)))$$