

Textures for Data Storage: Noise

CSCI 4239/5239

**Advanced Computer Graphics
Spring 2014**

Storing Data in Textures

- A texture can be thought of as a function
 - 1D => 1 independent variable
 - 2D => 2 independent variables
 - 3D => 3 independent variables
- Each texture is actually four functions
 - RGBA values are independent
 - Shader translates integer values to [0-1]
- Piecewise linear function description
 - User sets values at pixels
 - OpenGL interpolates between pixels

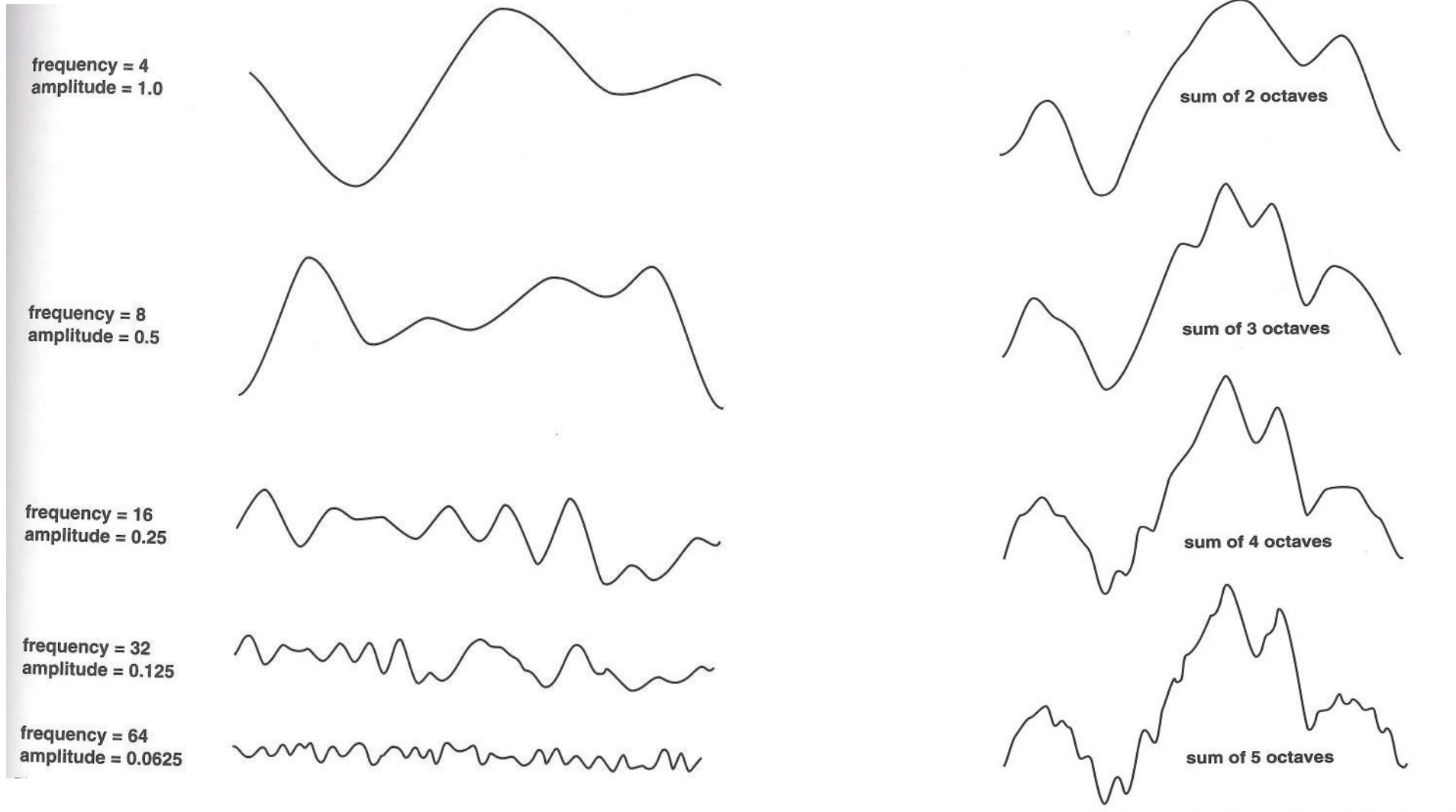
Applications

- Color translations
 - temperature to color
- Function approximation
 - sine, cosine
- Deformations/deviations
 - bump maps
- Distance
 - shadows
- Noise
 - parameters
 - results

Noise

- What is noise?
 - Random signals
 - Imperfections
 - Snoop Dogg
- Properties
 - Random (but not like random numbers)
 - Continuous
 - Range of frequencies

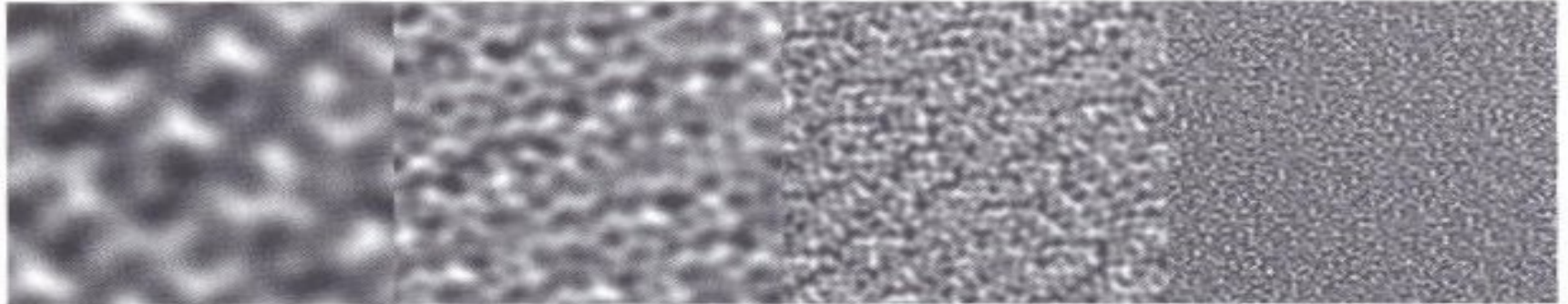
One Dimensional Noise



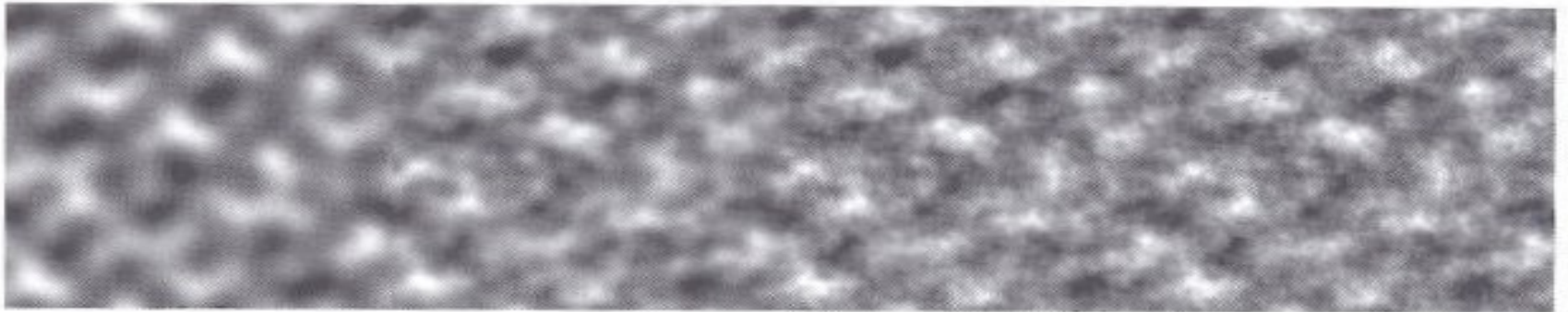
Components

Summation

Two Dimensional Noise



Components



Summation

Noise in Computer Graphics

- Introduced by Ken Perlin
 - First used in *Tron*
 - Academic paper in 1985
- Purpose
 - Rendering natural phenomena like clouds
 - Rendering materials like wood or concrete
 - Adding imperfections to surfaces like bumps
 - Adding imperfections to motion like jitters
- Noise and random numbers are related but quite different

Properties of noise

- Appear random
- Continuous
- Reproducible
- Well defined domain and range
- No obvious regularity (with some exceptions)
- Isotropic (with some exceptions)

Types of Noise

- Value noise
 - Values interpolated from points
- Gradient noise
 - Values generated from random gradients
 - Perlin (Classical) noise
 - Simplex noise
 - Faster and simpler than Perlin
 - Wavelet noise
 - More realistic

Noise in GLSL

- Part of the language
 - noiseX(vecY)
 - rarely implemented
 - no compiler error, just returns 0
- D-I-Y
 - Noise texture (ex17 and CSClx329)
 - Generates Perlin noise texture and sample in shader
 - Noise shader (ex18)
 - Store coefficients in texture
 - Compute Perlin and simplex noise in fragment shader

Results

