Noise!

Mike Bailey
mjb@cs.oregonstate.edu
Oregon State University

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License
Noise:

- Can be 1D, 2D, or 3D
- Is a function of input value(s)
- Ranges from -1. to +1. or from 0. to 1.
- Might look random, but really isn’t
- Has continuity
- Is repeatable (i.e., if you supply the same inputs, you will always get the same outputs)
**Positional Noise**

**Idea:** Pick a random number at the whole-number input values and then fit a piecewise smooth curve through those points.

The problem is that, due to the uncertainty of random numbers, you might get a good plus-or-minus distribution, or a not-so-good distribution.
Gradient Noise

**Idea:** Place points at the mid-line at the whole-number input values use random numbers to pick gradients (slopes) there, and then fit a piecewise smooth curve through those points with those slopes.

No matter what, you will get a good plus-or-minus distribution.
Quintic (5\textsuperscript{th} order) Interpolation Creates More Continuity Than Cubic

\textit{Cubic: }C^1 \textit{ continuity at the whole-number values} \quad \textit{Quintic: }C^2 \textit{ continuity at the whole-number values}
Coefficients for Cubic and Quintic Forms

\[ N(t) = C_{N0}N_0 + C_{N1}N_1 + C_{G0}G_0 + C_{G1}G_1 + C_{C0}C_0 + C_{C1}C_1 \]

**Cubic**

\[
\begin{align*}
C_{N0} &= 1 - 3t^2 + 2t^3 \\
C_{N1} &= 3t^2 - 2t^3 = 1 - C_{N0} \\
C_{G0} &= t - 2t^2 + t^3 \\
C_{G1} &= -t^2 + t^3 \\
C_{C0} &= 0 \\
C_{C1} &= 0
\end{align*}
\]

**Quintic**

\[
\begin{align*}
C_{N0} &= 1 - 10t^3 + 15t^4 - 6t^5 \\
C_{N1} &= 10t^3 - 15t^4 + 6t^5 = 1 - C_{N0} \\
C_{G0} &= t - 6t^3 + 8t^4 - 3t^5 \\
C_{G1} &= -4t^3 + 7t^4 - 3t^5 \\
C_{C0} &= \frac{1}{2}t^2 - \frac{3}{2}t^3 + \frac{3}{2}t^4 - \frac{1}{2}t^5 \\
C_{C1} &= \frac{1}{2}t^3 - t^4 + \frac{1}{2}t^5
\end{align*}
\]
Noise Octaves

**Idea:** Add multiple noise waves, each one twice the frequency and half the amplitude of the previous one.

1 Octave

4 Octaves
Image Representation of 2D Noise

1 Octave

4 Octaves
3D Surface Representation of 2D Noise

4 Octaves
3D Volume Rendering of 3D Noise

1 Octave

Has continuity in X, Y, and Z
3D Volume Isosurfaces of 3D Noise

1 Octave

S* = Mid-value

4 Octaves

The low half of the noise values are on side of the surface, the high half are on the other
Examples

Color Blending for Marble

Color Blending for Clouds

Deciding when to Discard for Erosion
Idea: Take the absolute value of the noise about the centerline, giving the noise a “sharper” appearance and creating “creases”. **Warning: this is not the same as fluid “turbulence”**.
Turbulence Example

Normal

Turbulent
How to Use Noise

Have an equation that relates some input value (x, y, z or u, v) to output values (color, height)

Have actual input values of where we are right now

Add Noise to the actual input values to produce new “fake” input values

Use those new “fake” input values in the original equation

Idea: The graphics system will display “here”, using display parameters as if you were “over there.
How to Use Noise

N = NoiseMag * noise( NoiseFreq * PP );

How much to amplify the noise effect
Coordinates where you are now
How much to increase the sampling rate

Should PP be in Model or World coordinates? Why?
\[ N = \text{NoiseMag} \cdot \text{noise}(\text{NoiseFreq} \cdot \text{PP}); \]
Surface Shader Only
Displacement Shader Only
Surface and Displacement Shaders together
No Noise

Surface Only

Displacement Only

Surface + Displacement

Noise

Surface Only

Displacement Only

Surface + Displacement
What’s the Difference Between These Two RenderMan Images? Why?

Displacement-mapped

P = P + normalize(N) * disp;
N = calculatenormal(P);

Bump-mapped

N = calculatenormal( P + normalize(N) * disp );
What’s the Difference Between These Two RenderMan Images? Why?

```plaintext
P = P + normalize(N) * disp;
N = calculatenormal(P);
```

```plaintext
if( disp != 0. )
{
    P = P + normalize(N) * disp;
    N = calculatenormal(P);
}
```