AntiAliasing Perlin Noise

Presented at Siggraph 2001

Ian Stephenson

National Centre for Computer Animation
Bournemouth University
Poole, Dorset, BH5 12BB
United Kingdom
istephen@bournemouth.ac.uk

1 Introduction

The Perlin Noise function is a key tool in procedural texturing, providing the controlled randomness required to create visual interest. However, as with all periodic functions, it is prone to aliasing when sampled at a frequency below the Nyquist frequency. The main approaches used to limit these artifacts are super-sampling, and frequency clamping, at the cost of render time, and shader complexity respectively.

The preferred solution to any aliasing problem is to convolve the continuous signal with a sampling kernel. If we accept a simple box filter, then this reduces to calculating the integral of the signal over the sampling area. However the random nature of noise, and a lack of understanding of its implementation has led many shader writers to believe that this is not viable.

This sketch will demonstrate that the integration of noise is relatively simple provided that lattice gradients are available.

THIS FIGURE CURRENTLY UNAVAILABLE

Figure 1: standard noise(left) Vs inoise (right)

2 The 1D Perlin Noise Function

Perlin Noise is a form of gradient lattice noise. A random gradient is generated for each integer lattice point (where the value of the noise is zero), and the gradients are smoothly interpolated between these values.

Interpolation is performed through the use of wavelets. These wavelets are defined to be zero outside the range ± 1, and integrate to zero over that range. When integrating 1D noise it is therefore only necessary to consider the two wavelets which intersect the sample point. These wavelets are simple polynomials, and hence may be trivially integrated. In forming the definite integral these must be evaluated at each end of the sampling area, and hence integrated noise is approximately twice as expensive as point sampling.

3 The Two Dimensional case

In two dimensions, the wavelets are arranged in a grid. Once again wavelets fully outside or fully inside the area being integrated over will sum to zero, and may be safely ignored. Those wavelets which intersect the corners of the area may be integrated in a similar fashion to the 1D case (at a total cost approximately four times that of point sampling). However in the 2D case we now need to consider those wavelets which intersect the edges of the area (or the faces of the volume in higher dimensional cases). Fortunately, the wavelet functions are of a form which allows the integration of an entire edge to be evaluated as a simple summation of gradients, multiplied by a single polynomial. The computational cost of these edge integrals can therefore be kept to a minimum.

This approach generalizes to higher dimensions, and though the cost of the boundry evaluation increases with both dimension, and size of the summed area, it does so more slowly than supersampling.

4 Implementation

The integrated noise (Inoise) functions have been implemented, both in a custom renderer, and as DSO shadeops for Pixar's PRMan. A nave implementation of the maths is numerically unstable due to the mixing of floating point operations with the division of space into lattice cells. However a stable implementation has been developed, and has been shown to accurately reproduce the standard noise function while greatly reduced aliasing artifacts.