SCI INSTITUTE

TECHNICAL REPORT

Better Gradient Noise

Andrew Kensler, Aaron Knoll and Peter Shirley

UUSCI-2008-001

Scientific Computing and Imaging Institute
University of Utah
Salt Lake City, UT 84112 USA

April 8, 2008

Abstract:

We present three improvements to Perlin's gradient noise algorithm. First, a small change in the permutation hash function combined with separate pseudorandom tables yields significantly better axial decorrelation. Second, a modification to the reconstruction kernel approximating a global higher-order differencing operator produces better bandlimitation. Third, the quality of 2D surfaces using solid 3D noise is improved by reconstructing the stencil projected onto a surface normal. These three techniques are mutually orthogonal, generalize to higher dimensions, and are applicable to nearly any gradient noise, including simplex noise. Combining them yields a desirable Fourier spectrum for graphics applications.

THE UNIVERSITY OF UTAH

Better Gradient Noise

Andrew Kensler
SCI Institute, University of Utah

Aaron Knoll
SCI Institute, University of Utah

Peter Shirley
NVIDIA Corporation

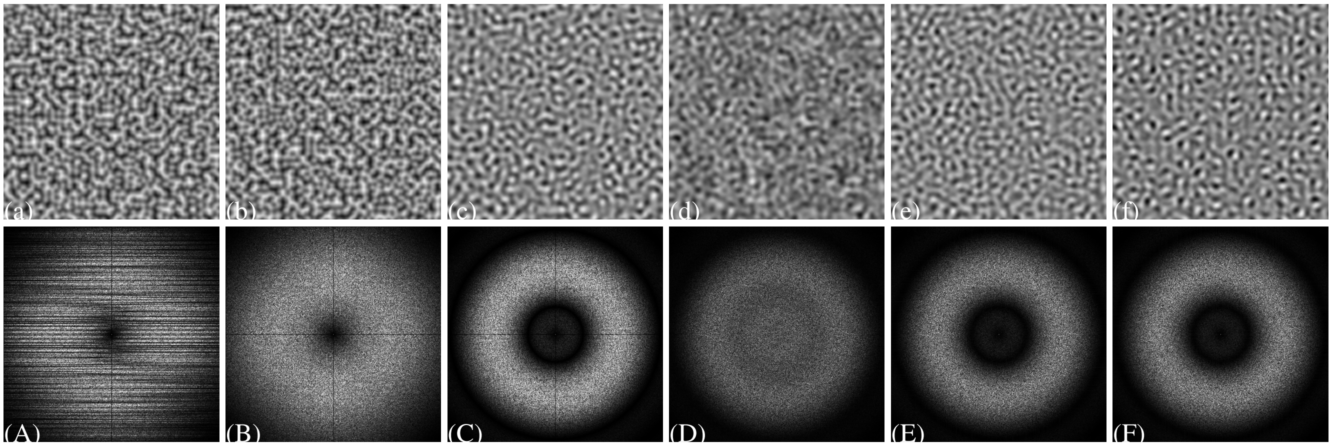

Figure 1: *Detail of images and associated Fourier transforms of (a) 2D Perlin noise with standard hash function, quintic interpolant and 256-entry gradient table, (b) 2D Perlin noise with our* xor *hash function from Section 2, (c) 2D noise with our extended reconstruction kernel from Section 3, (d) Slice of our 3D noise along plane normal to $x = y = z$, (e) Same planar slice with our projection method from Section 4, (f) Same planar slice and projection method with faster half-resolution grid. All 2D transforms computed from full repetitions.*

Abstract

We present three improvements to Perlin's gradient noise algorithm. First, a small change in the permutation hash function combined with separate pseudorandom tables yields significantly better axial decorrelation. Second, a modification to the reconstruction kernel approximating a global higher-order differencing operator produces better bandlimitation. Third, the quality of 2D surfaces using solid 3D noise is improved by reconstructing the stencil projected onto a surface normal. These three techniques are mutually orthogonal, generalize to higher dimensions, and are applicable to nearly any gradient noise, including simplex noise. Combining them yields a desirable Fourier spectrum for graphics applications.

Keywords: noise, procedural texture, shading

1 Introduction

The Perlin gradient noise [1989] function is a key component in most rendering systems. Ideal synthetic noise should exhibit no discernible periodicity, anisotropy or aliasing. This implies a Fourier spectrum that is bandlimited and rotationally invariant. The original implementation employs differenced separable cubic Hermite filters sampling from a regular grid, and exhibits noticeable anisotropy. To remedy this, Perlin's simplex noise [2001] samples from a simplicial grid and uses a radial reconstruction filter. Perlin [2002] improved the results from the cubic grid and by using a higher-degree polynomial separable convolution filter. While improvements, the latter work does not remove the axial correlation responsible for most of the anisotropy, and neither significantly attenuates lower frequencies. An alternative to gradient noise, wavelet noise [Cook and DeRose 2005] avoids the axially-correlated periodicity, and ensures a desirable bandlimited spectrum. However, it displays anisotropy in the lower bandlimit which affects the visual appearance when viewed closely. In this paper, we first show how to remove axial correlation in standard Perlin noise. We then present a modification that improves bandlimits as in wavelet noise, while preserving the visual characteristics of the classic Perlin noise. Finally, we show how to take 2D projections of 3D Perlin noise to achieve desirable visual properties.

2 Hash Function

Perlin [1989] generates random unit vectors on the 3D integer lattice indexed as i, j, k. To achieve finite storage, a single table P of N randomly permuted integers $0 \ldots (N-1)$ is hashed successively using each axial coordinate, $H_{ijk} = P[P[P[i]+j]+k]$, where indices are assumed to be modulo the table length. This provides the index into another table $\mathbf{G}$, containing unit-length gradient vectors.

The purpose of the hash is to decorrelate the indices. However, if i and j are held constant and steps are taken along k, this will unfortunately produce successive entries in P. For any values of $\{i, j, k\}$ that hash to $P[0]$, the adjacent lattice point $\{i, j, k+1\}$ will always produce $P[1]$. In fact, each column will produce exactly the same sequence of hash values as any other – the copies will simply be shifted. This breaks the fundamental assumption that the samples of the random noise process are uncorrelated. Given a sufficiently large sampling of Perlin noise, the spectrum manifests strong striations perpendicular to the preferred axis, shown in Fig. 1A.

To fix this, we use a separate permutation table for each dimension, P_x, P_y, P_z, and take the exclusive-or of the value from each:

$$H_{ijk} = P_x[i] \oplus P_y[j] \oplus P_z[k]$$

Fig. 1B shows the improvement to the spectrum that this produces. While requiring a slight increase in memory, this has the advantage of eliminating dependent permutation table lookups and reducing the total number of those lookups from 14 to 6 in the case of 3D Perlin noise. In the case of ND noise, the number of lookups is reduced from $2^{N+1} - 2$ to $2N$.

3 Filter Kernel

Though our hashing scheme spectrum fixes most of the anisotropy of the Fourier spectrum, the bandlimits are quite weak: both low and high frequencies remain. Provided that the random noise process is truly uncorrelated, the overall shape of the frequency spectrum is determined by the reconstruction kernel.

When considered in one dimension, Perlin noise uses an antisymmetric kernel of the form $s(x)x$ where $s(x)$ is either an Hermite cubic polynomial [Perlin and Hoffert 1989] or a quintic polynomial [Perlin 2002]. The latter produces a pair of lobes with opposite signs that peak at a distance of approximately 0.80 from each other and then fall to zero beyond that.

We note that the separable Perlin filter very closely resembles a continuous (smoothed) version of the discrete impulse response of a first-order (forward or backward) difference operator. Moreover, successive application of a difference operator k times to a given impulse causes an impulse response of the $k+1$ set of binomial coefficients with alternating signs. Applied as a filter, this scales frequency f by $(2\sin(\pi f))^k$ [Hamming 1998]. Effectively, differencing attenuates the lower third of the Nyquist interval and amplifies the upper two-thirds, explaining the strong presence of high frequencies and the near-linear attenuation of low frequencies in the Perlin noise spectrum. The smoothing isolates the first replica and eliminates the higher order harmonics (Fig. 1B).

Higher order differencing ensures better bandlimits, but at the cost of a wider filter. The lowest k for which the stencil encloses the next immediately neighboring samples is $k = 3$, a 4-point stencil on $[-2, 2]$. The binomial coefficients of third-order forward differencing are 1,-3,3,-1, so we desire additional opposing lobes one unit away and 1/3 the amplitude of the inner pair. For computational efficiency our filter approximates this as a polynomial $s(x)$; which is again applied to the antisymmetric Perlin noise kernel of the form $s(x)x$. In choosing $s(x)$ we seek a symmetric (even-degree) polynomial that makes $s(x)x$ satisfy the previous constraint on the lobes, and has $s' = s'' = 0$ at the stencil endpoints. Searching this family of functions suggests the following:

$$\begin{aligned} s(x) &= (2-x)^4(2+x)^4(1-x)(1+x)/256 \\ &= 4(1-x^2/4)^5 - 3(1-x^2/4)^4 \end{aligned}$$

We implement the function in the second form for efficiency. The remaining question is whether this polynomial adequately approximates the corrected differencing filter. Using a separable radial filter of this function in 2D produces the results shown in Fig. 1c, in which high frequencies are effectively attenuated. The disadvantage of our method is that the 4-point stencil is costly, requiring 16 lookups in 2D and 64 in 3D. However, this larger kernel allows contributions from the additional grid points to eliminate the regular zeroes of both classic Perlin noise and simplex noise, and is clearly effective in producing a band-pass spectrum.

4 Projection to 2D

In their use as solid textures, 3D noises are frequently sampled along 2D surfaces. However, Cook and DeRose [2005] showed that even if a noise is 3D bandlimited, a planar slice will not be bandlimited due to the consequences of the Fourier slice theorem. Instead, low frequencies will be present and the Fourier transform will appear as a solid disk (Fig. 1D). To solve this problem, they project the wavelet noise onto a surface by performing a weighted line integral along the surface normal under the assumption that the curvature is weak at the scale of the noise. Inspired by their method, we propose a simple projection technique applicable to gradient noises.

Our method projects each of the neighboring points for which the kernel is evaluated onto the plane tangent to the surface, and evaluates the kernel directly at those projected points. Because of the radial nature of the kernel this results in a planar slice using a convolution kernel equivalent to the 2D kernel. To compensate for the projection and to avoid popping, we weight the contribution from each neighboring point with a cubic Hermite curve that falls off with the distance above or below the plane. Fig. 1e shows our result: low frequencies are attenuated isotropically, yielding a more radially symmetric spectrum than wavelet noise.

The following pseudocode employs our hash, filter, and projection method (given unit-length surface normal $\mathbf{N}$). Indices into the permutation tables P_x, P_y, P_z are assumed to be modulo table size. Table $\mathbf{G}$ contains vectors uniformly distributed on the unit sphere.

```
function ProjectedNoise(X, N)
    I ← {⌊X_x⌋, ⌊X_y⌋, ⌊X_z⌋}
    F ← X − I
    v ← 0
    for k ← −1 to 2 do
        for j ← −1 to 2 do
            for i ← −1 to 2 do
                D ← F − {i, j, k}          ▷ vector from lattice point
                o ← D · N                  ▷ offset from plane
                A ← D − oN                 ▷ projection onto plane
                d ← A · A                  ▷ squared distance to projection
                o ← 1 − |o|
                if d < 4 and o > 0 then
                    h ← P_x[I_x + i] ⊕ P_y[I_y + j] ⊕ P_z[I_z + k]
                    t ← 1 − d/4
                    v ← v + (A · G[h])(4t^5 − 3t^4)(3o^2 − 2o^3)
    return v
end
```

5 Discussion

We have shown how to improve the spectral properties of Perlin noise with relatively minor changes. The hash method of Section 2 improves both visual quality and runtime (about 5%) with only a modest increase in storage; we believe all standard noise implementations should adopt it. The kernel and projection method yield a bandlimited spectrum attenuating high and low frequencies, respectively. These methods are more expensive to evaluate (about $6\times$ and $14\times$ respectively), thus their use should be restricted to applications where the visual benefits merit the penalty. While we have focused on visual quality, we note that with our method we can trade some visual appearance for speed and still retain the isotropic bandlimits by sampling the grid at half-resolution—i.e., using only the integer lattice points with even-valued coordinates (Fig. 1f). This variant is well suited to summations over multiple octaves and effectively returns to a 2-point stencil at an evaluation cost of only $2.4\times$ with projection. In either case, the fewer dependent lookups needed for the new hashing scheme implies opportunities for optimization.

Acknowledgements

We wish to thank Steven Parker, Erik Brunvand and the other members of the Utah Hardware Ray Tracing project for many useful discussions. This work was supported by NSF grant 05-41009.

References

COOK, R. L., AND DEROSE, T. 2005. Wavelet noise. *ACM Trans. Graph. 24*, 3, 803–811.

HAMMING, R. W. 1998. *Digital Filters*, third ed. Dover Publications, Inc.

PERLIN, K., AND HOFFERT, E. 1989. Hypertexture. In *SIGGRAPH*, 253–262.

PERLIN, K. 2001. Noise hardware. In *Real-Time Shading SIGGRAPH Course Notes*, M. Olano, Ed. ch. 9.

PERLIN, K. 2002. Improving noise. In *SIGGRAPH*, 681–682.