DSO simplex noise

I have written a DSO shadeop for the RenderMan Interface which implements simplex noise as a new SL function simplexnoise(). It's free software, and you can download it below.

- Download the Linux version
- Download the Windows version

(Both archives have identical content except for the precompiled binary)

If you have some other platform than Linux or Windows, you need to compile the DSO yourself for your platform, but it should compile and run straight off with most RenderMan-compatible renderers out there. Even for Windows or Linux, you *might* have to replace the "shadeop.h" file with the corresponding one for your renderer and recompile. The particular header file I use here is from Aqsis, and the pre-compiled DSO (the .dll file for Windows, the .so file for Linux) worked fine for me with Aqsis, RenderDotC, BMRT and PRMan. Other renderers might want a recompilation of the DSO.

Why use simplex noise?

Classic noise has problems with non-uniformity throughout its domain of definition, particularly for 2D planar slices of 3D and 4D noise, it has visible axis-aligned artefacts, it is expensive to compute for 4D and up, and its derivative in 3D and 4D is a very complicated high order polynomial.

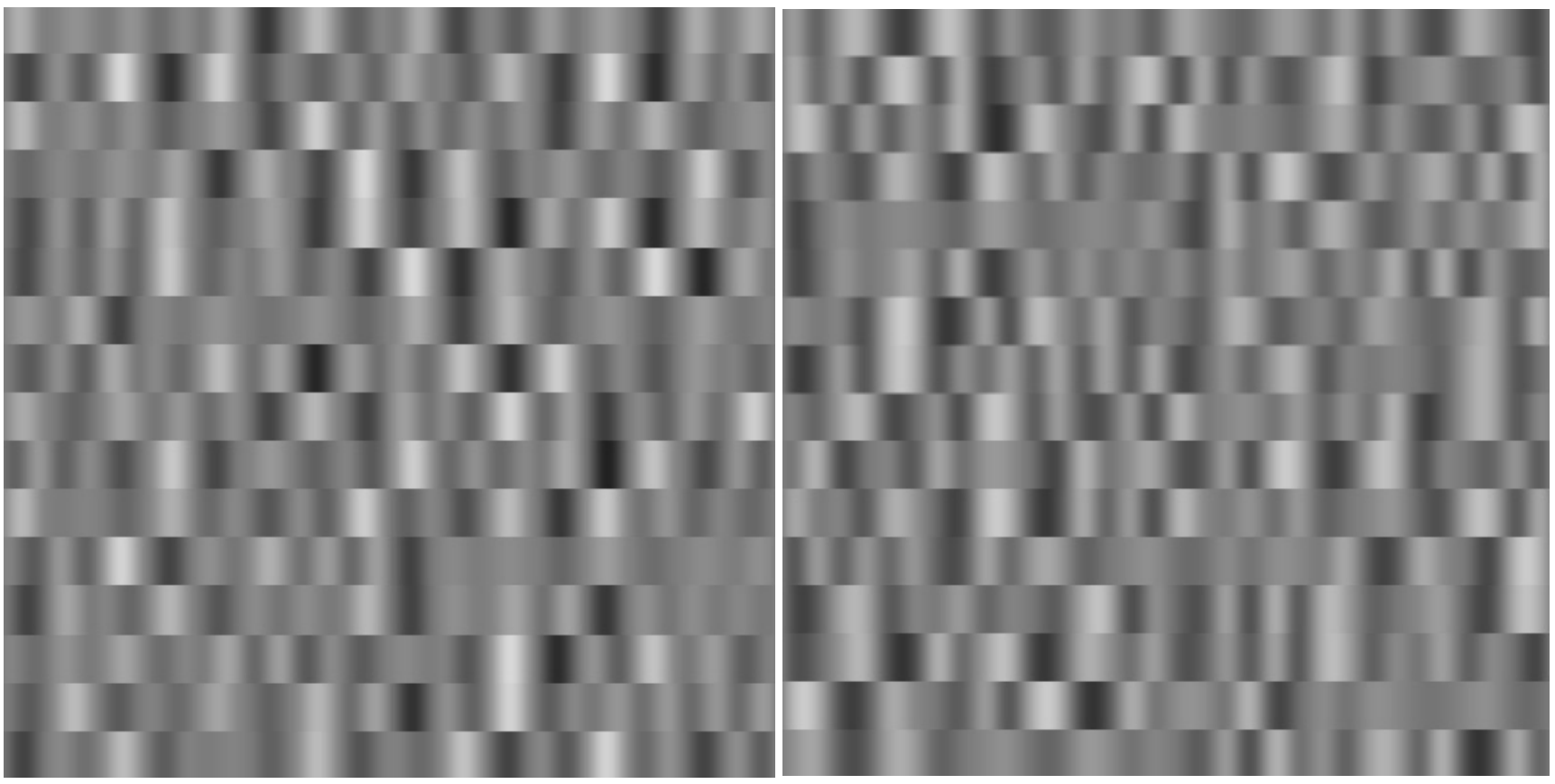
Simplex noise is several times faster to compute, particularly for 4D and up, it does not have nearly as many visual problems with non-uniformity and axis-aligned artefacts, and it has a simple polynomial derivative everywhere, even for higher dimensions.

Simplex noise looks better, *but* different, and is thus visually incompatible with classic Perlin noise. The difference in feature size and range of values can easily be compensated for by a few simple scaling multiplications, but the different visual character might change the visual result of shaders that depend heavily on one or two components of noise. (Fractal sums of several noise components should still look about the same, though.)

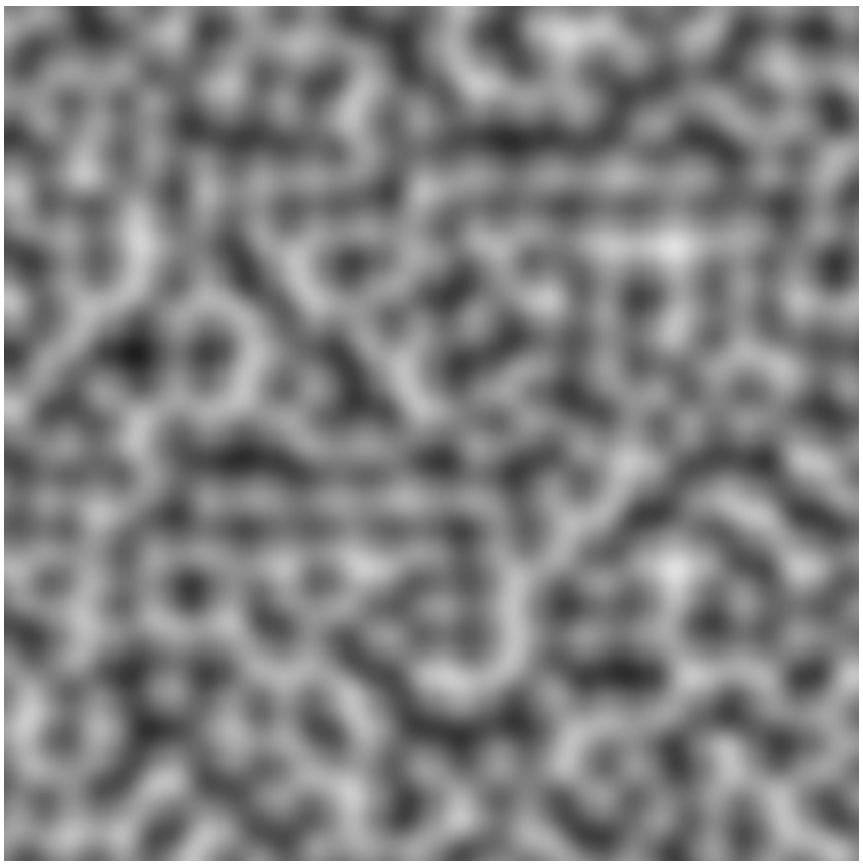
Visual comparison of classic Perlin noise to simplex noise

Perlin Improved noise (from Aqsis 1.1) | **Perlin simplex noise (from my DSO)**

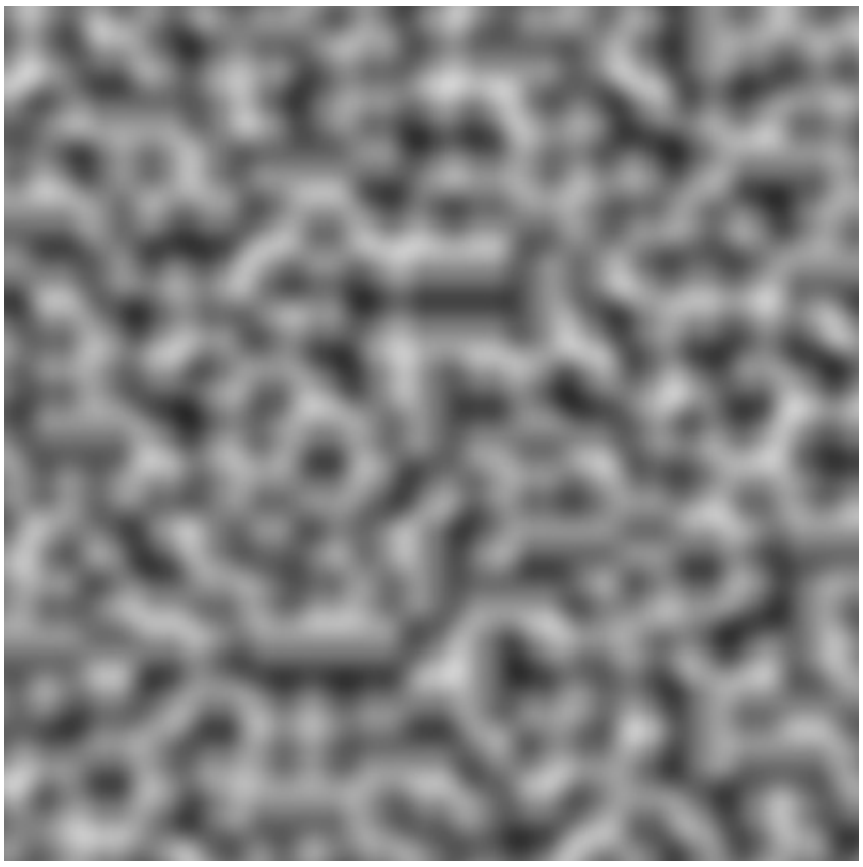
noise(u)



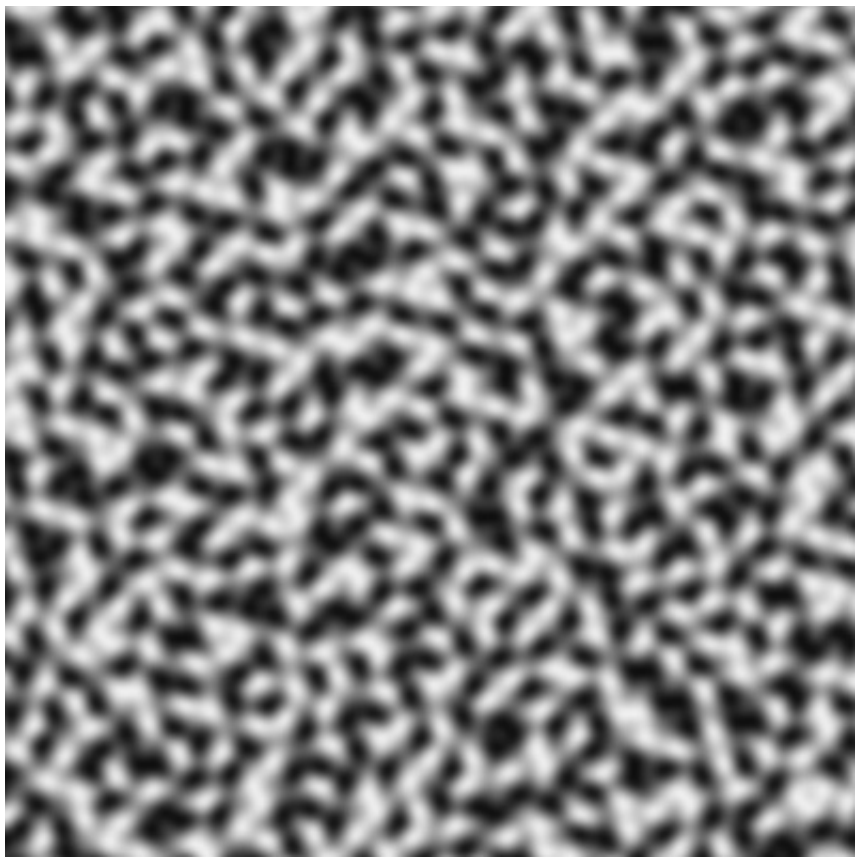
noise(u,v)



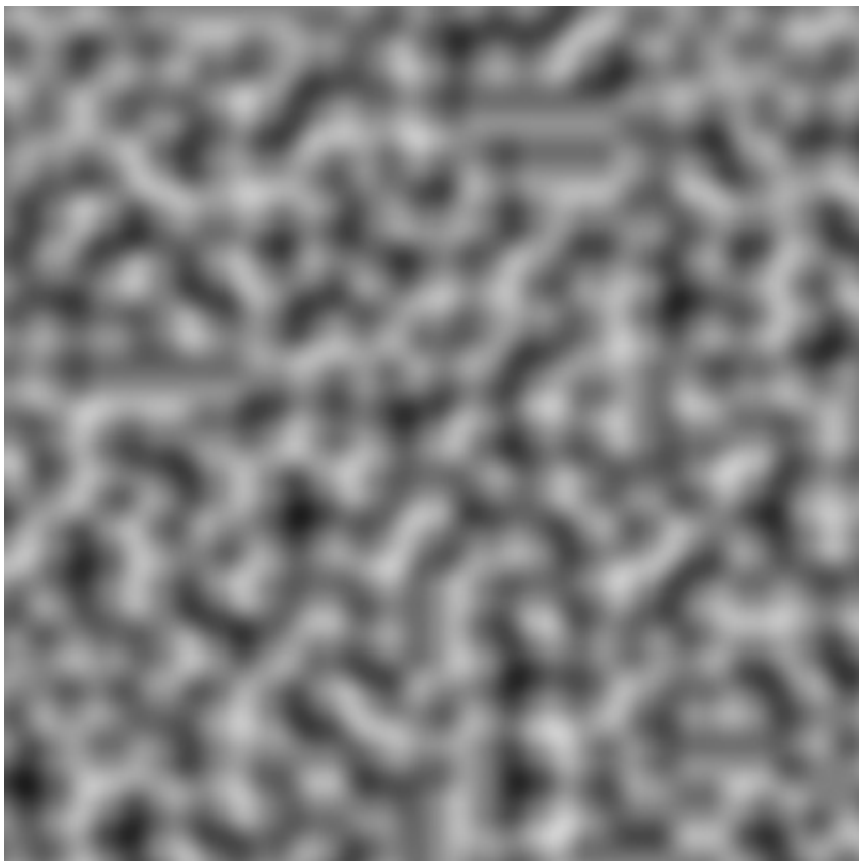
noise(x,y,z)

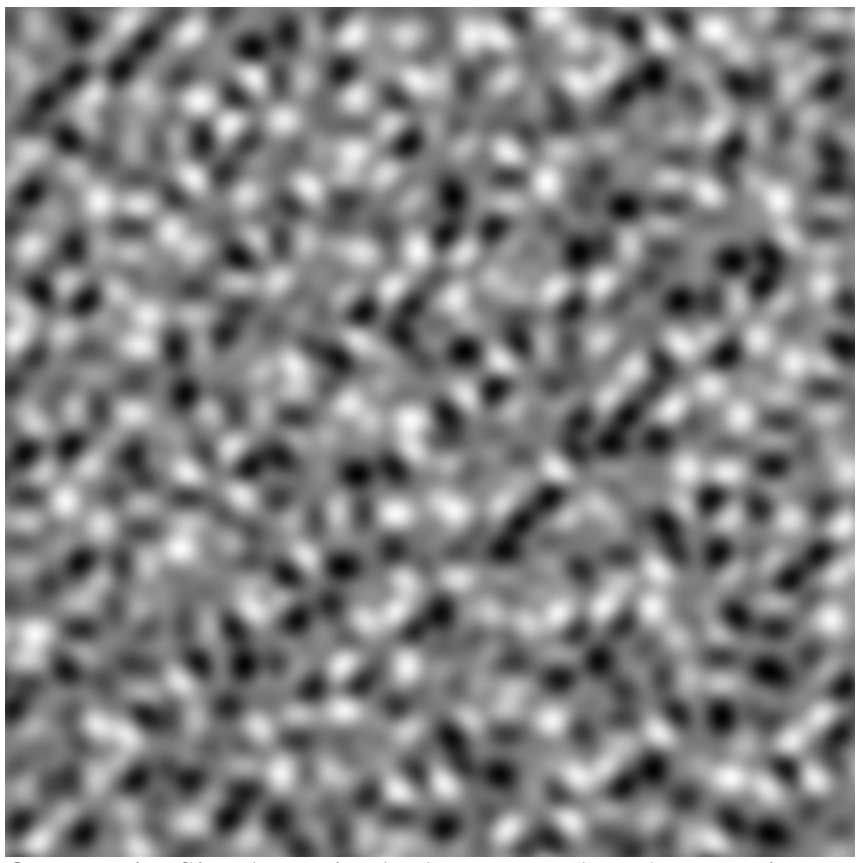


noise(x,y,z,t)



Simplex noise looks more or less the same in any planar slice, contrary to classic noise which looks quite different near integer values in z than half-way between two integers in z.





Once again, Simplex noise looks more or less the same in any planar slice, but classic noise looks very much different depending on whether z or t, or both, are near integer values or not.

Stefan Gustavson 2005-11-28