Perlin Noise

CS-116B: Computer Graphics Algorithms

Perlin noise: history

- Ken Perlin wrote a paper for 1985's SIGGRAPH (Special Interest Group on Graphics) conference titled, "An Image Synthesizer".
- In this paper, Ken Perlin proposed the use of statistical noise generator to simulate texture on a computer graphics surface.
- This paper has since become a seminal work in the area of Computer Graphics

Perlin noise: advantages

- The Perlin noise function has the following properties:
  - “Statistical invariance under rotation (no matter how we rotate its domain, it has the same statistical character)” (p. 289).
  - “A narrow bandpass limit in frequency (its [sic] has no visible features larger or smaller than within a certain narrow size range)” (p. 289).
  - “Statistical invariance under translation (no matter how we translate its domain, it has the same statistical character)” (p. 289).

Source: Perlin, K. (1985). *An Image Synthesizer*. Proceedings of the 12th annual conference on Computer graphics and interactive techniques, San Francisco, CA (pp. 287-296). New York: ACM. doi:10.1145/325334.325247

Perlin noise is high performance

- Perlin noise functions are finite and discrete: “…the calculation stops at the pixel level. In this way unwanted higher frequencies are automatically clamped” (p. 292).
- Perlin noise functions can be computed concurrently: “…the algorithm proceeds independently at all sample points” (p. 292).

Source: Perlin, K. (1985). *An Image Synthesizer*. Proceedings of the 12th annual conference on Computer graphics and interactive techniques, San Francisco, CA (pp. 287-296). New York: ACM. doi:10.1145/325334.325247

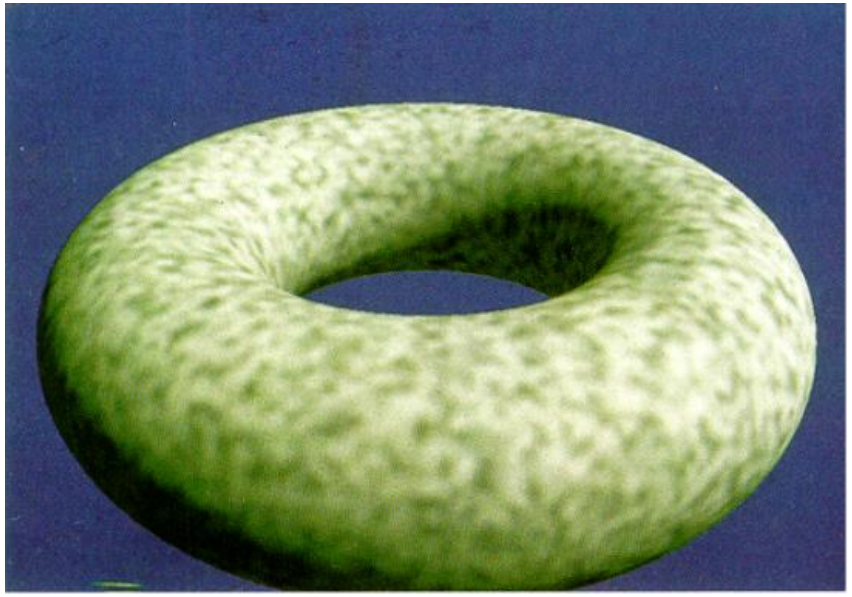
Perlin noise: simulations

Perlin noise can also be used to simulate the following:

- Fire
- Water
- Marble patterns

Source: Perlin, K. (1985). *An Image Synthesizer*. Proceedings of the 12th annual conference on Computer graphics and interactive techniques, San Francisco, CA (pp. 287-296). New York: ACM. doi:10.1145/325334.325247

Perlin noise example: “Spotted Donut”



*color = white * Noise(point)*

“The above texture has a band-limited character to it; there is no detail outside of a certain range of size. This is equivalent to saying that the texture's frequency spectrum falls off away from some central peak frequency” (p. 289).

Source: Perlin, K. (1985). *An Image Synthesizer*. Proceedings of the 12th annual conference on Computer graphics and interactive techniques, San Francisco, CA (pp. 287-296). New York: ACM. doi:10.1145/325334.325247

Perlin noise example: “Bozo's Donut”

*color = Colorful(Noise(k * point))*

"Through functional composition we may do many different things with the value returned by the Noise() function. For example, we might wish to map different ranges of values into different colors" (p. 289).

Source: Perlin, K. (1985). *An Image Synthesizer*. Proceedings of the 12th annual conference on Computer graphics and interactive techniques, San Francisco, CA (pp. 287-296). New York: ACM. doi:10.1145/325334.325247

Perlin noise example: “Bumpy Donut”

normal += Dnoise(point)

"Another convenient primitive is the vector valued differential of the *Noise()* signal, defined by the instantaneous rate of change of *Noise()* along the x, y, and z directions, respectively. We will call this function *Dnoise()*. *Dnoise()* provides a simple way of specifying normal perturbation." (pp. 289-290).

Source: Perlin, K. (1985). *An Image Synthesizer*. Proceedings of the 12th annual conference on Computer graphics and interactive techniques, San Francisco, CA (pp. 287-296). New York: ACM. doi:10.1145/325334.325247

Perlin noise example: marble vase

Source: Perlin, K. (1985). *An Image Synthesizer*. Proceedings of the 12th annual conference on Computer graphics and interactive techniques, San Francisco, CA (pp. 287-296). New York: ACM. doi:10.1145/325334.325247

For additional reading…

Bevins, J. (2007). *Libnoise*. http://libnoise.sourceforge.net/

How to Use Perlin Noise in Your Games. (2009). http://devmag.org.za/2009/04/25/perlin-noise/

Perlin, K. (1985). *An Image Synthesizer*. Proceedings of the 12th annual conference on Computer graphics and interactive techniques, San Francisco, CA (pp. 287-296). New York: ACM. doi:10.1145/325334.325247

Perlin, K. (2002). *Improving noise*. Proceedings of the 29th annual conference on Computer graphics and interactive techniques, San Antonio, TX (pp. 681-682). New York: ACM. doi:10.1145/566570.566636

Understanding Perlin Noise. (2014). http://flafla2.github.io/2014/08/09/perlinnoise.html