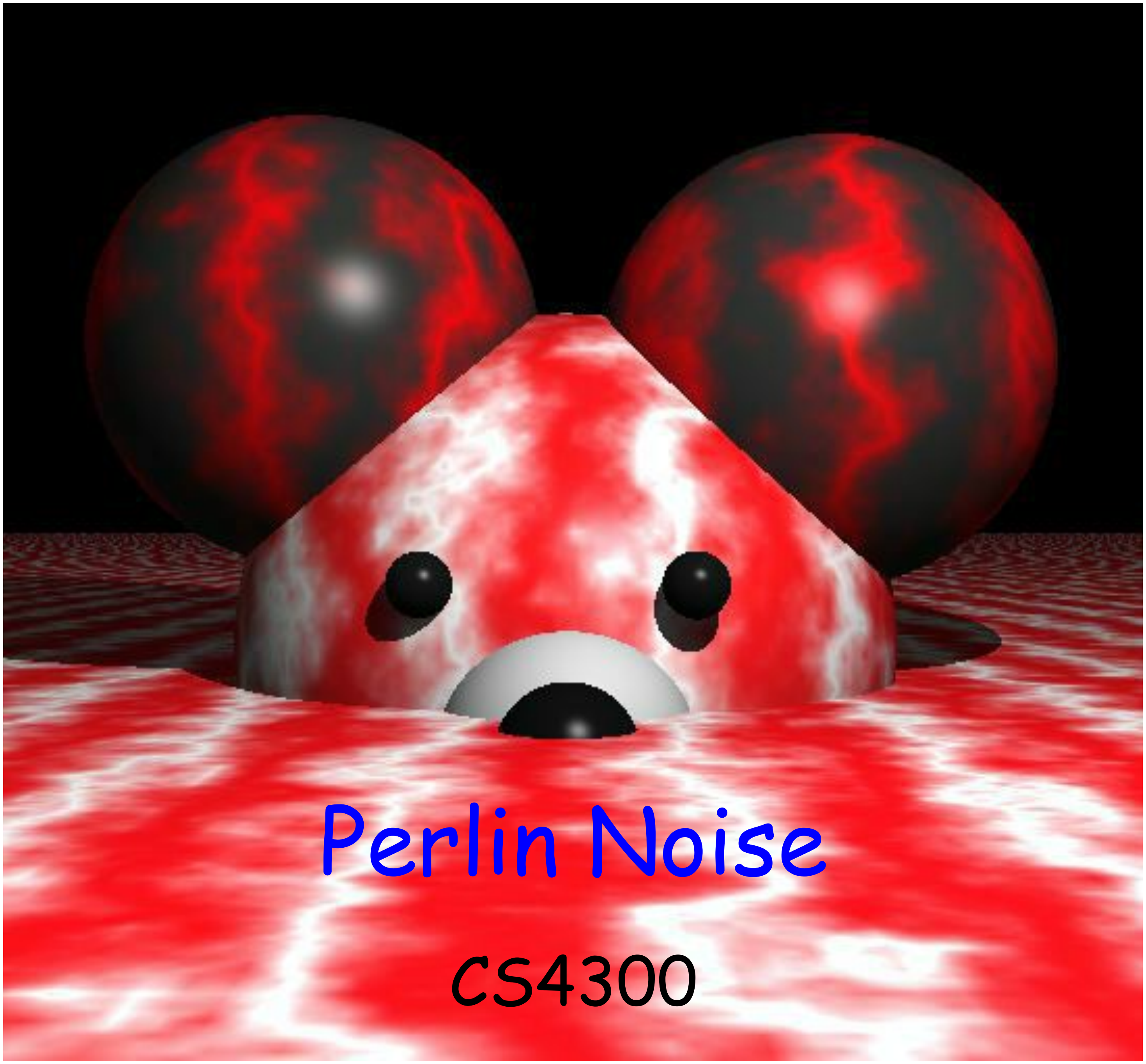


Perlin Noise

CS4300

The Oscar™

To Ken Perlin for the development of Perlin Noise, a technique used to produce natural appearing textures on computer generated surfaces for motion picture visual effects.

The Movies

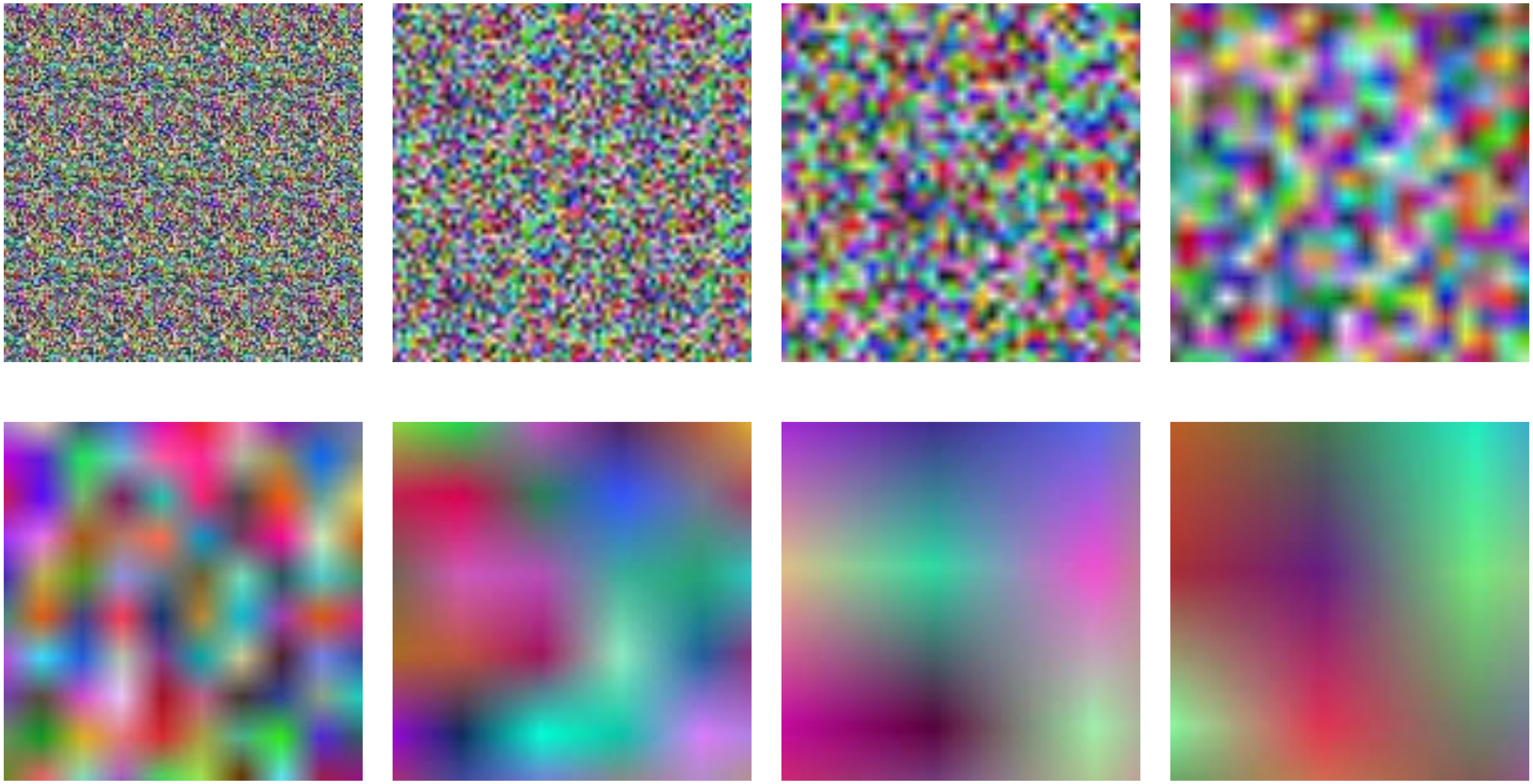
- James Cameron Movies (Abyss,Titanic,...)
- Animated Movies (Lion King, Moses,...)
- Arnold Movies (T2, True Lies, ...)
- Star Wars Episode I
- Star Trek Movies
- Batman Movies
- *and lots of others*

In fact, after around 1990 or so, *every* Hollywood effects film has used it.

What is Noise?

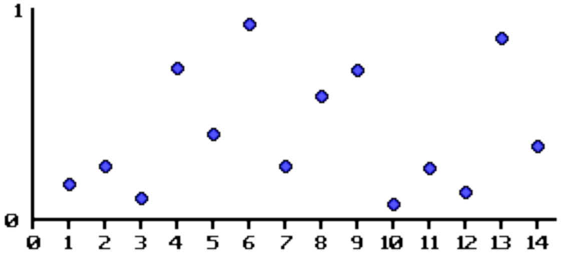
- Noise is a mapping from R^n to R - you input an n-dimensional point with real coordinates, and it returns a real value.
- n=1 for animation
- n=2 cheap texture hacks
- n=3 less-cheap texture hacks
- n=4 time-varying solid textures

Noise is Smooth Randomness

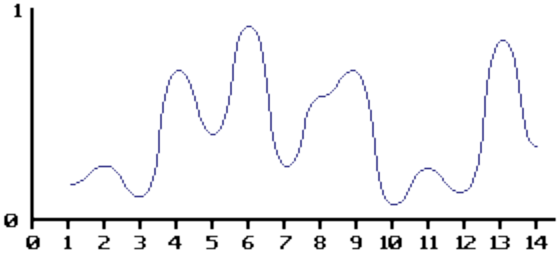


Making Noise

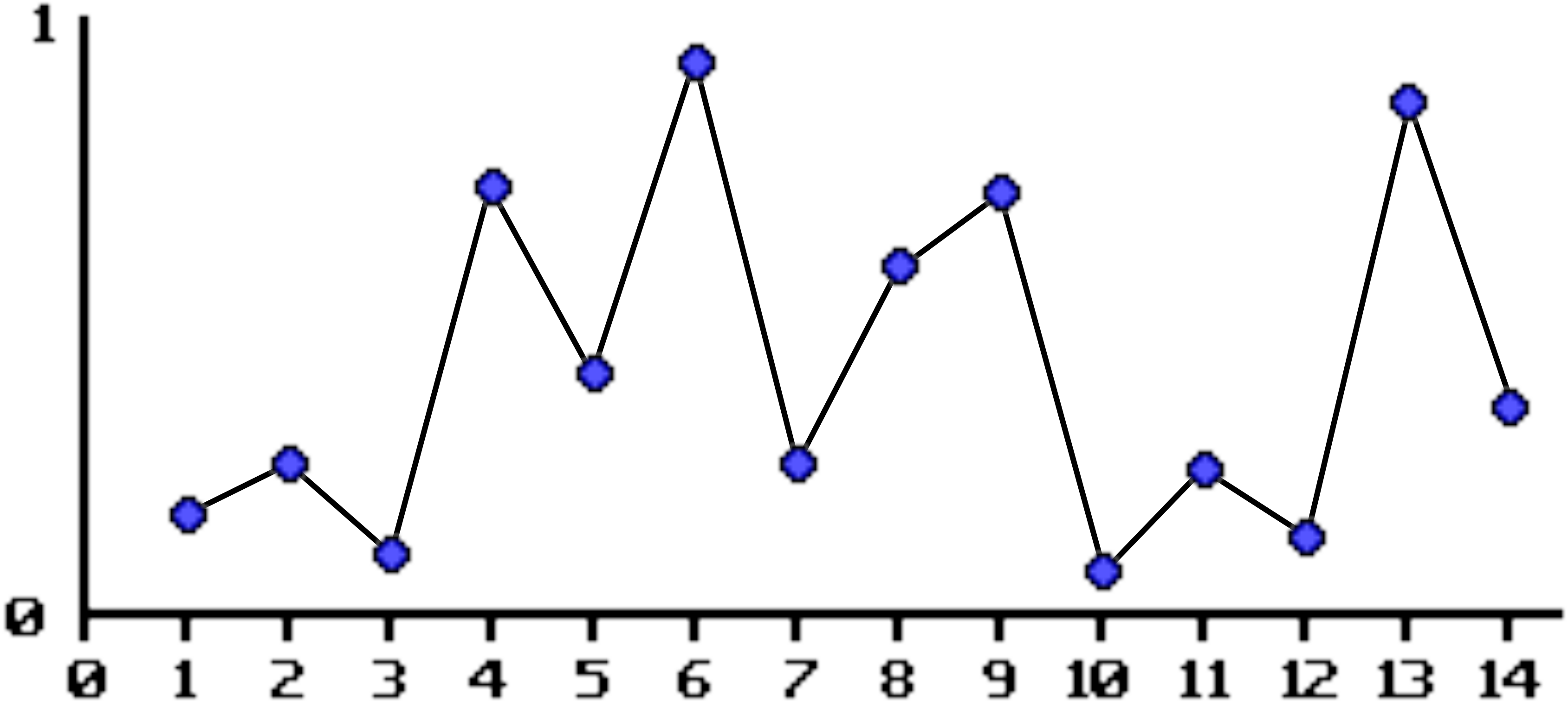
1. Generate random values at grid points.



2. Interpolate smoothly between these values.



Linear Noise

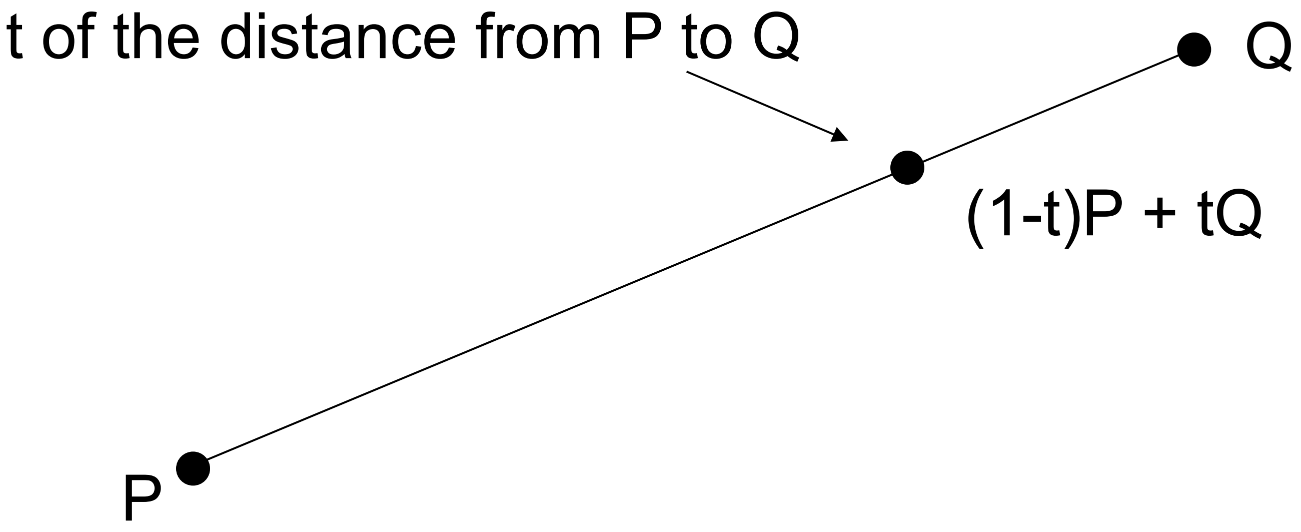


lerp

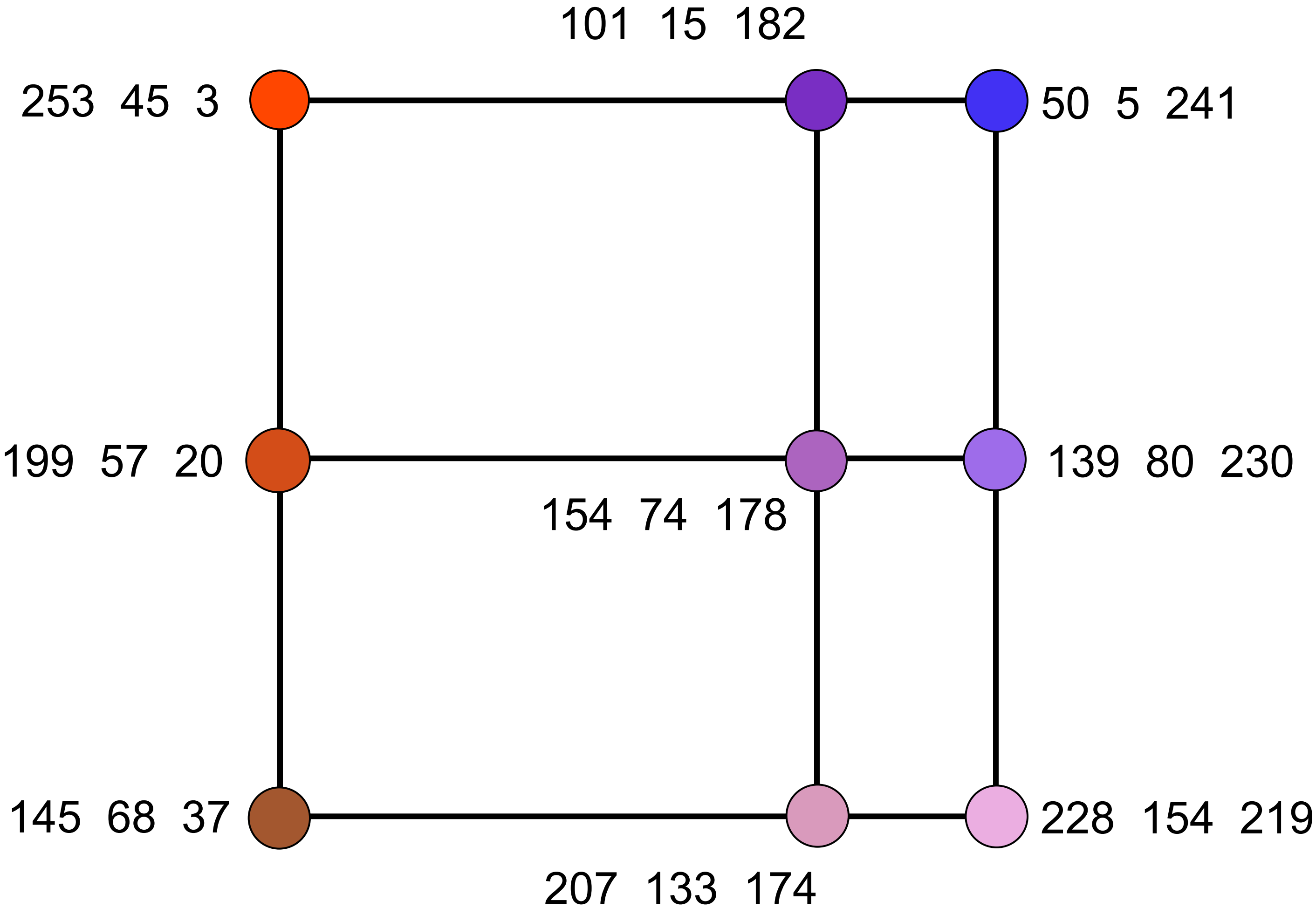
- The basic operation of linear interpolation between two values is so commonly used in computer graphics that it is sometimes called a ***lerp*** in the jargon of computer graphics.
- Lerp operations are built into the hardware of all modern computer graphics processors.

lerping

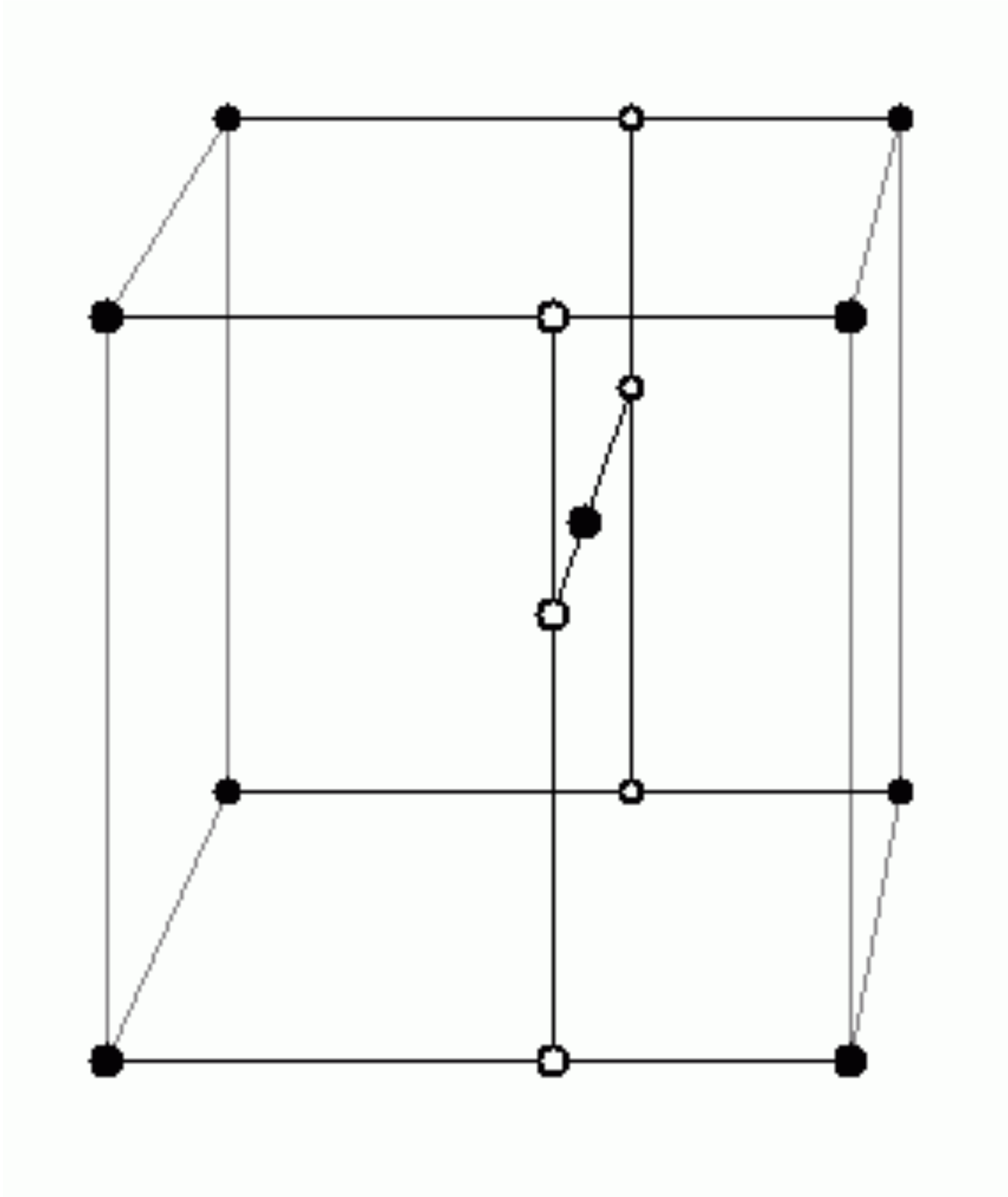
$$\mathrm{lerp}(v1, v2, t) = (1 - t)v1 + tv2$$



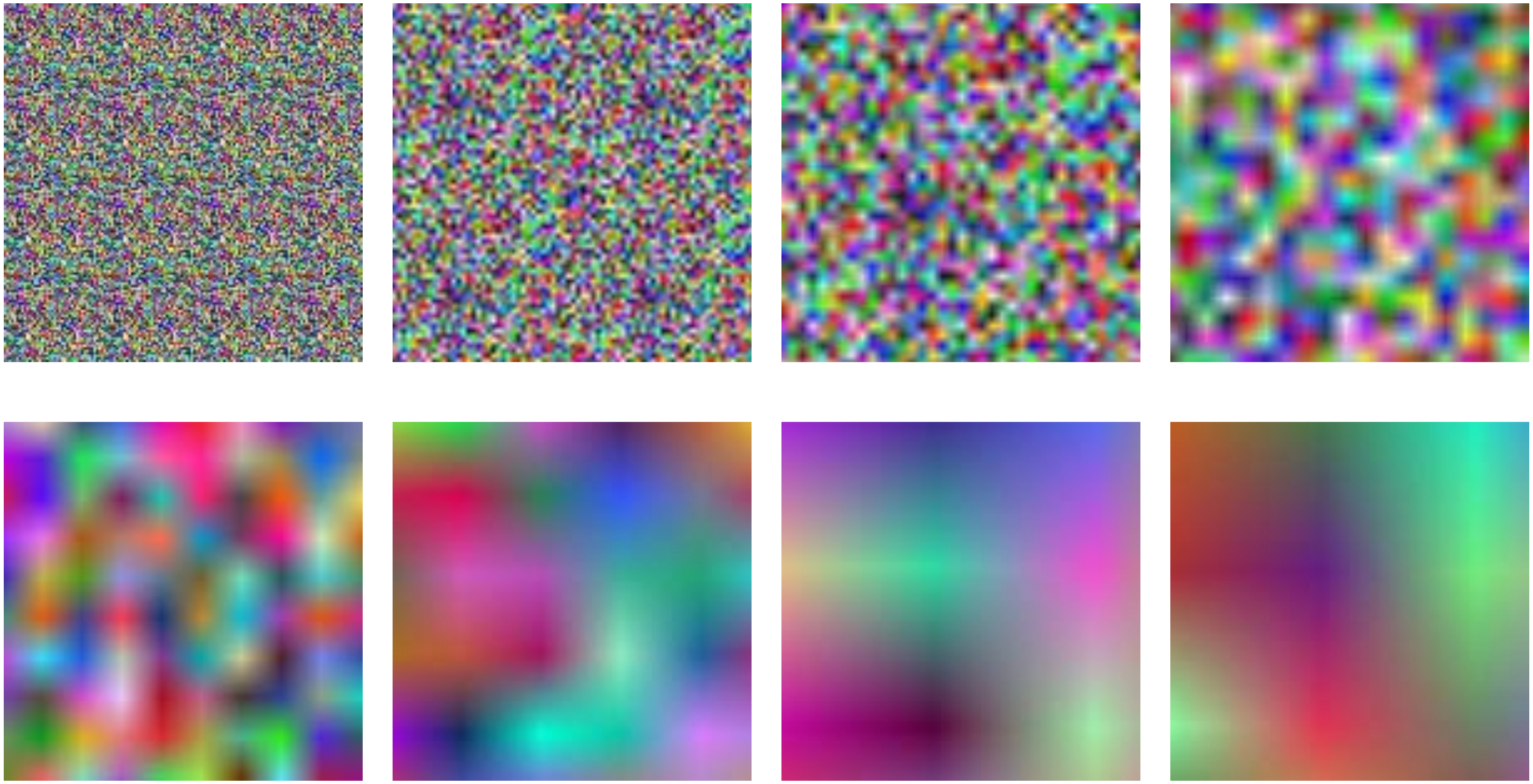
2D Linear Noise



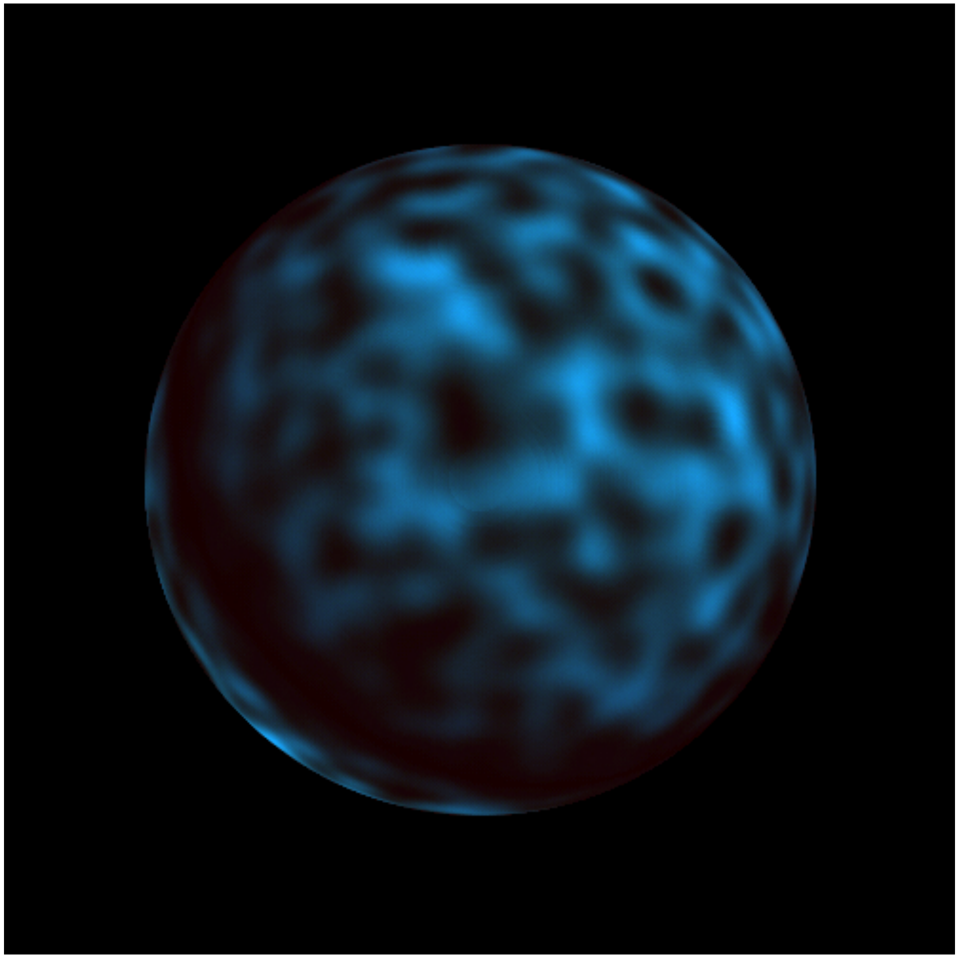
3D Linear Noise



Noise is Smooth Randomness

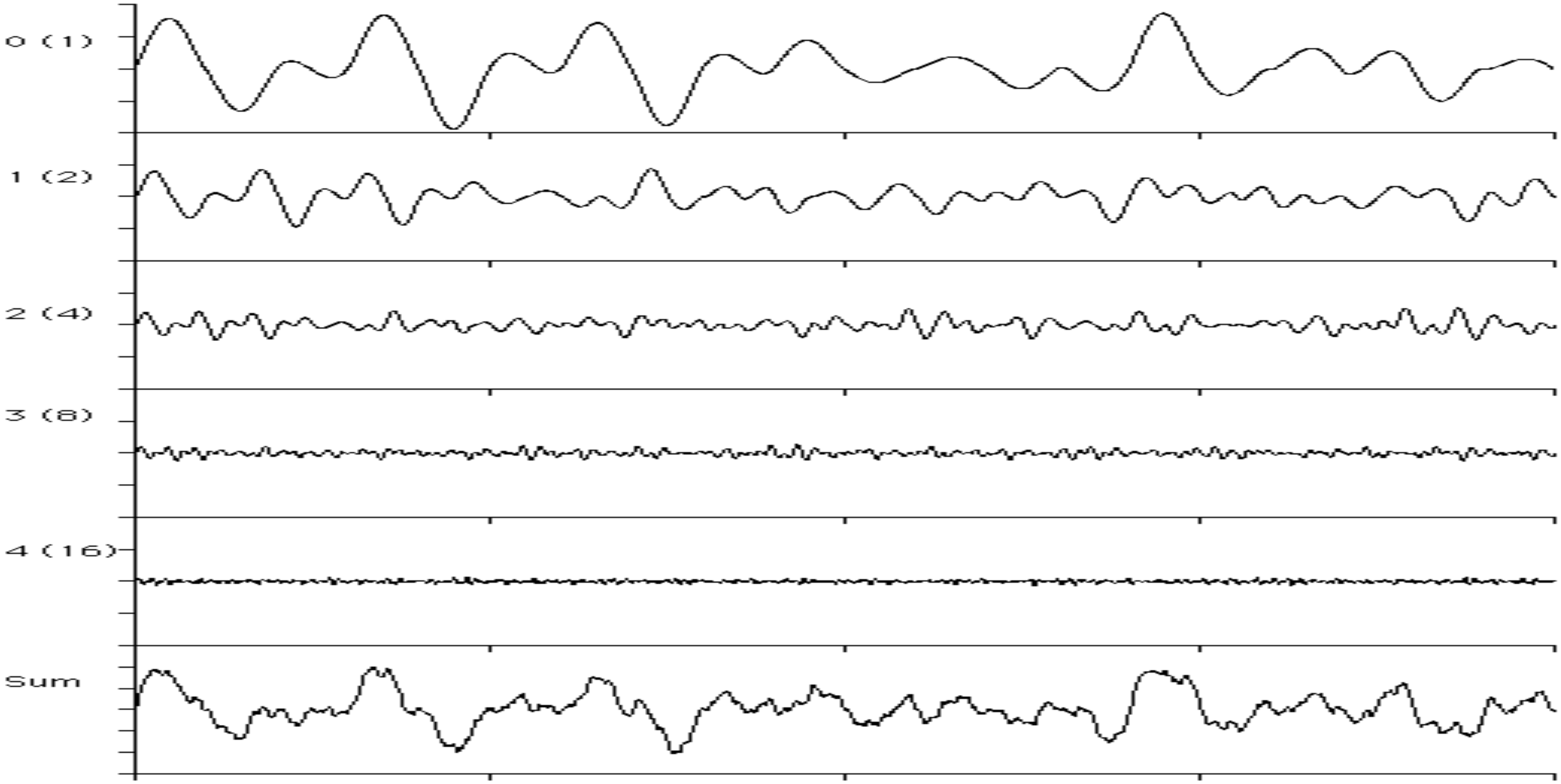


Perlin Noise Sphere

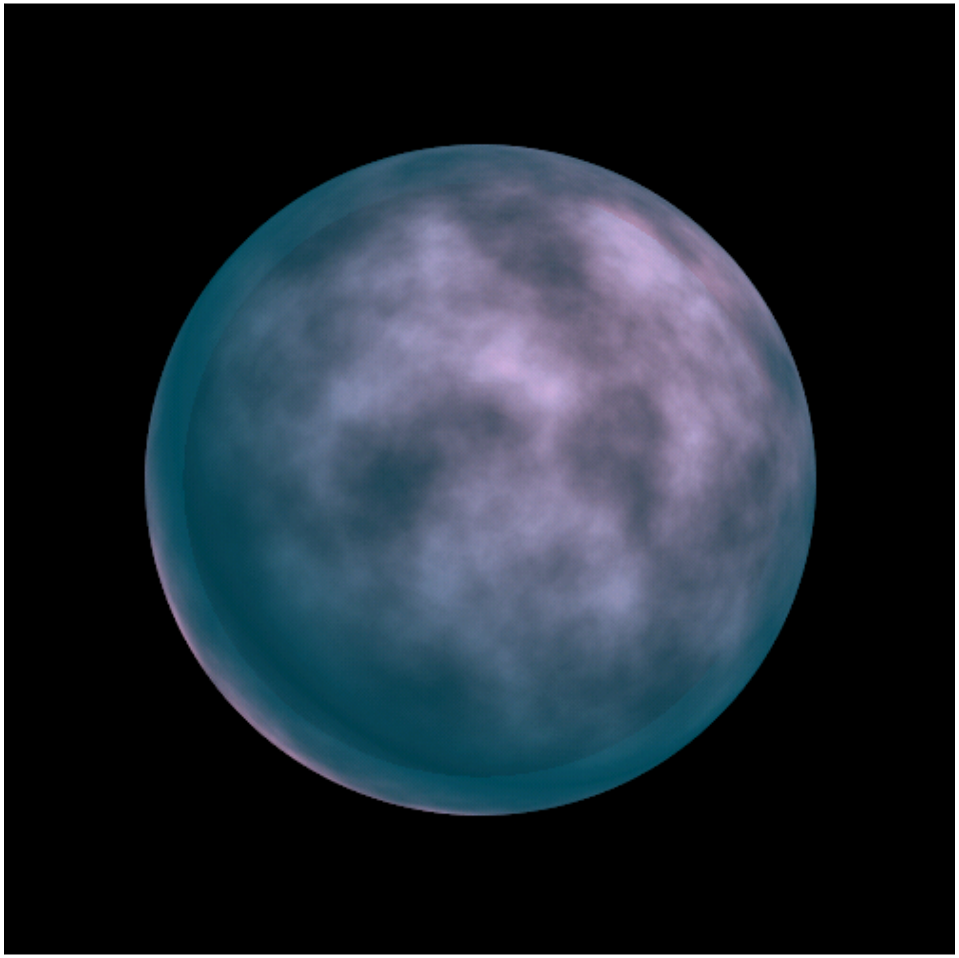


Turbulence or Sum 1/f(noise)

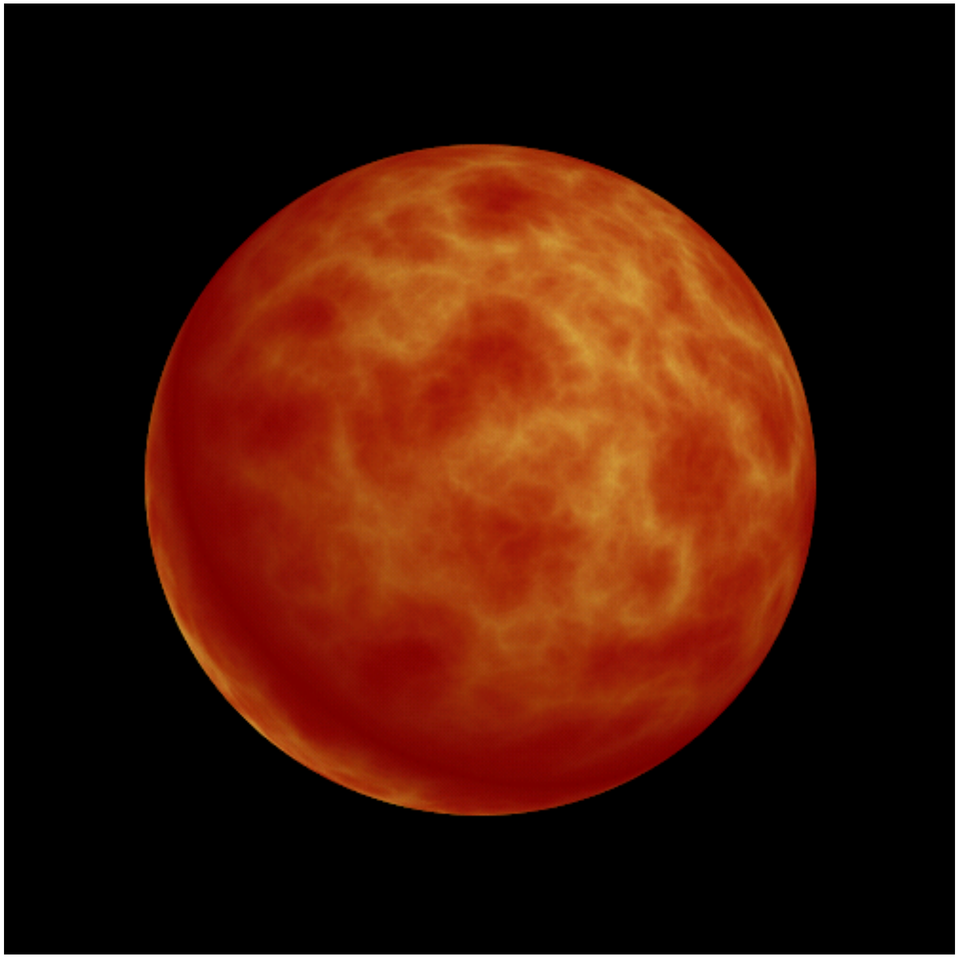
$\text{noise}(\mathbf{p}) + \frac{1}{2}\,\text{noise}(2\mathbf{p}) + \frac{1}{4}\,\text{noise}(4\mathbf{p})\ \ldots$



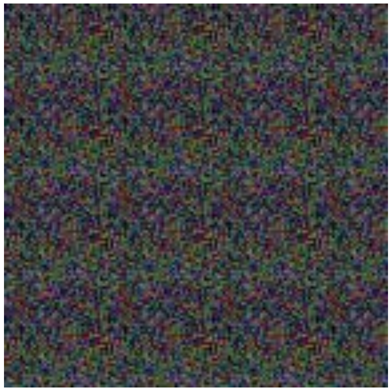
Perlin Sum 1/f(noise) Sphere

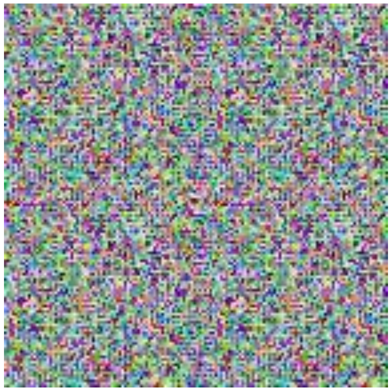


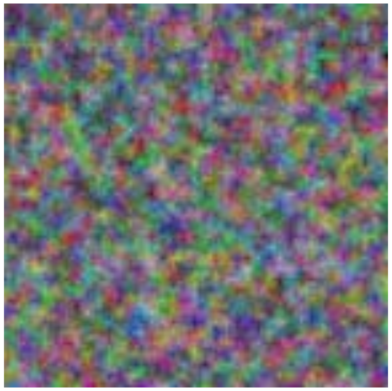
Perlin Sum 1/f(|noise|) Sphere

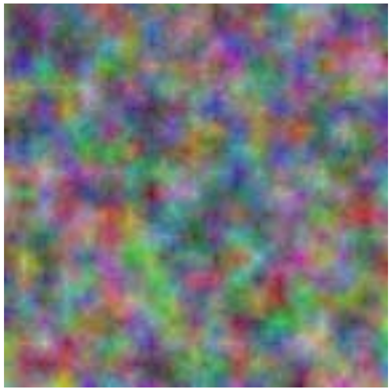


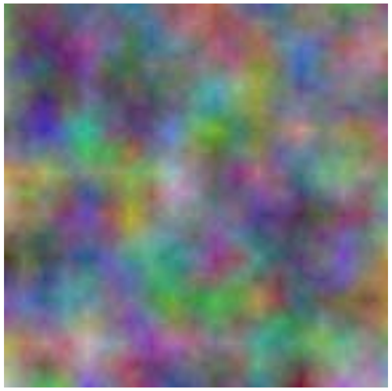
2D Nornalized Turbulence

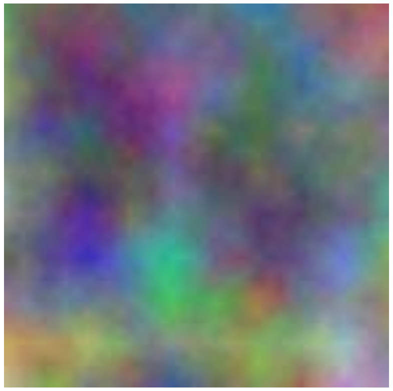


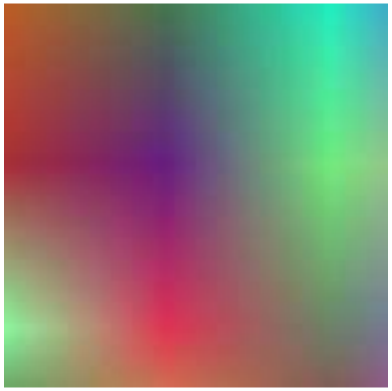






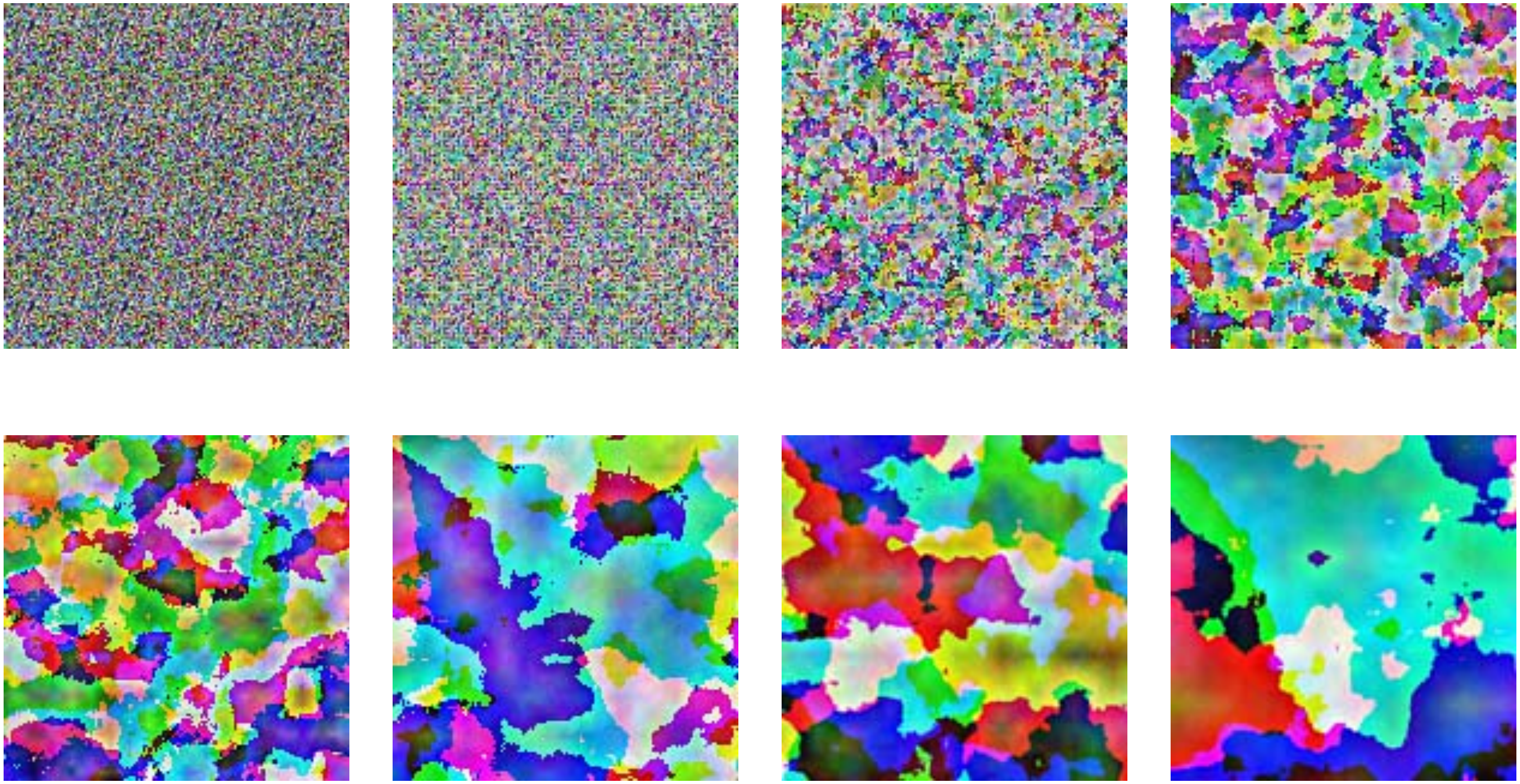




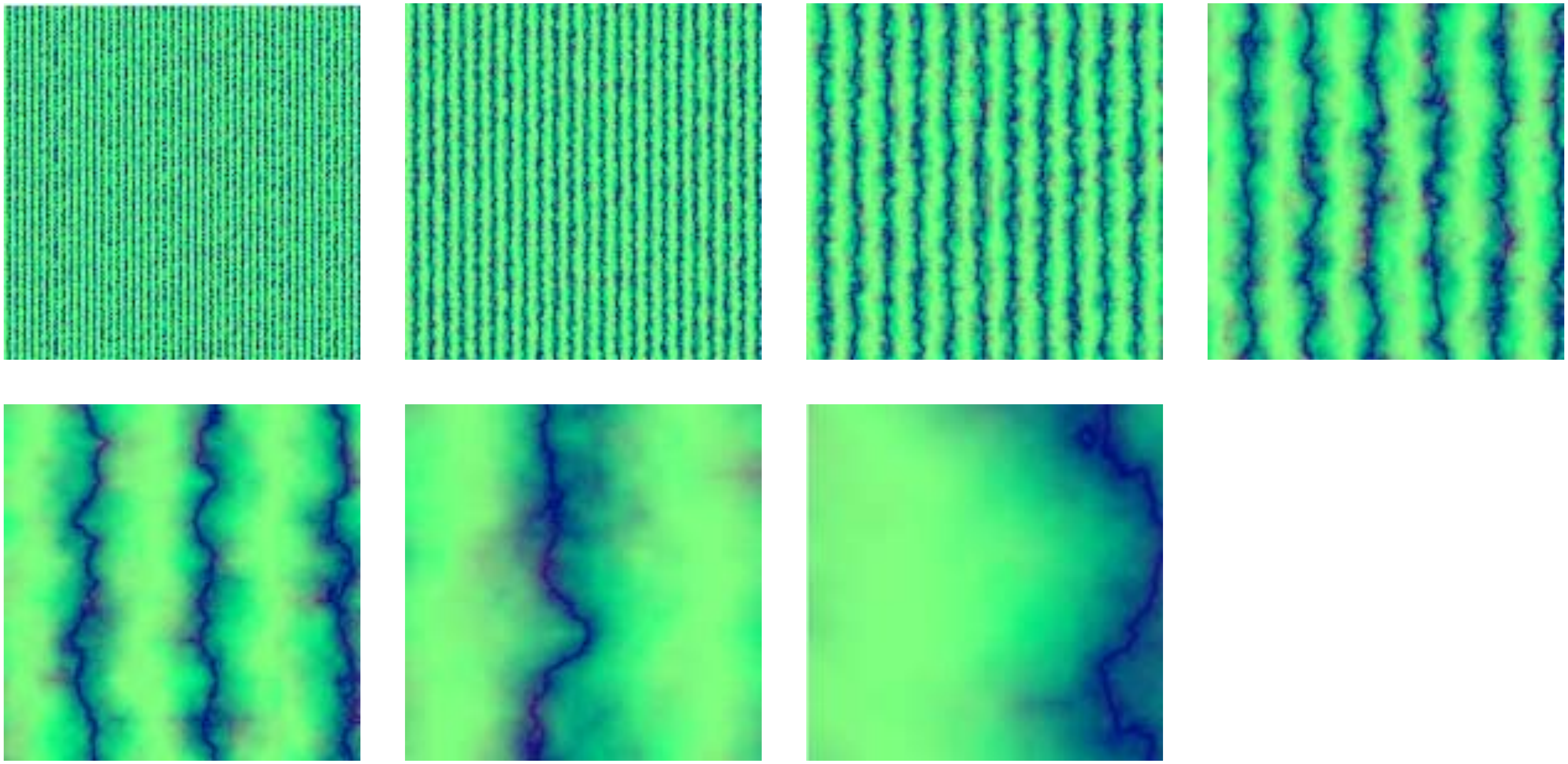


Just Noise

2D Turbulence - Clipped

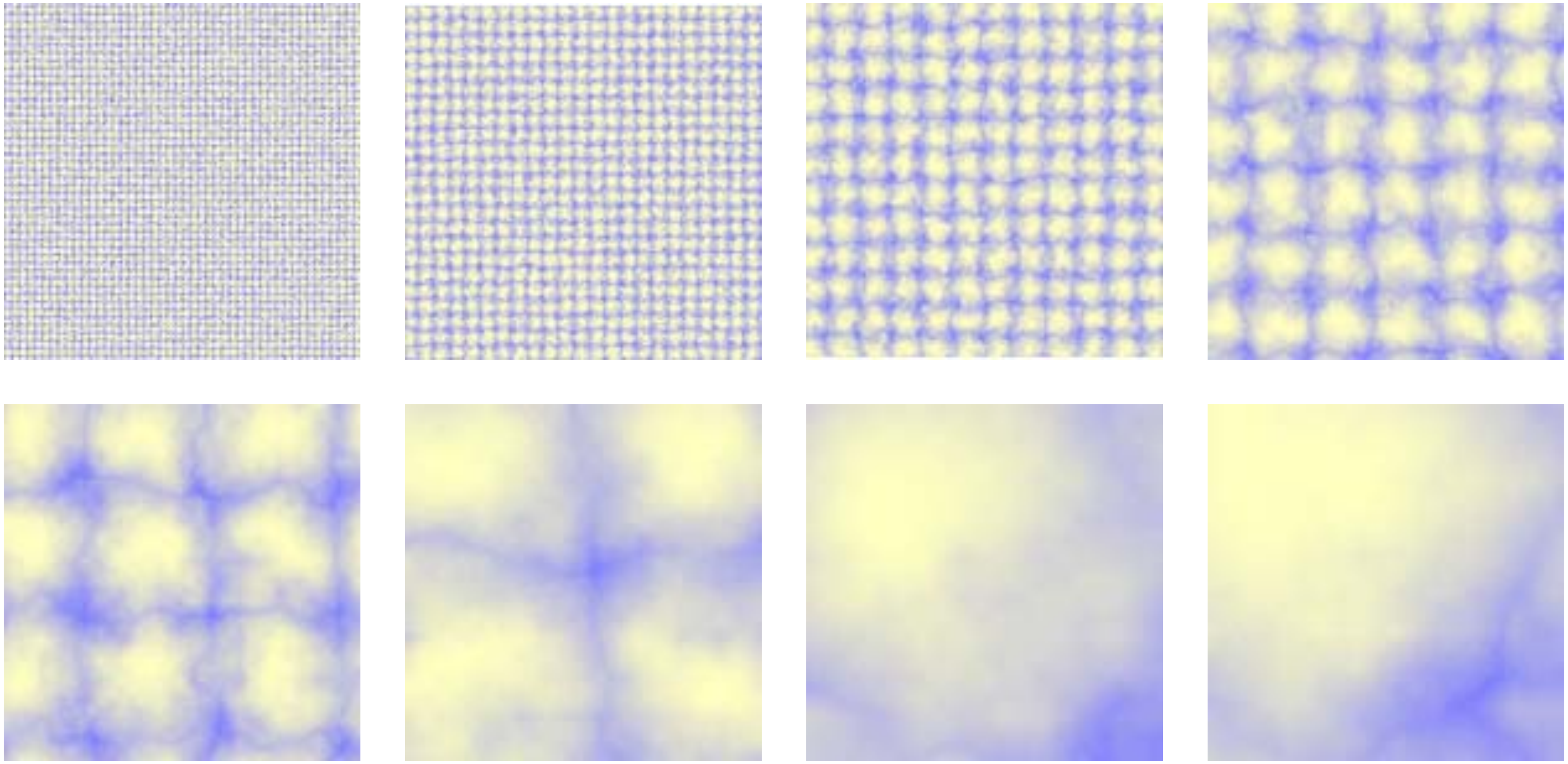


Marble



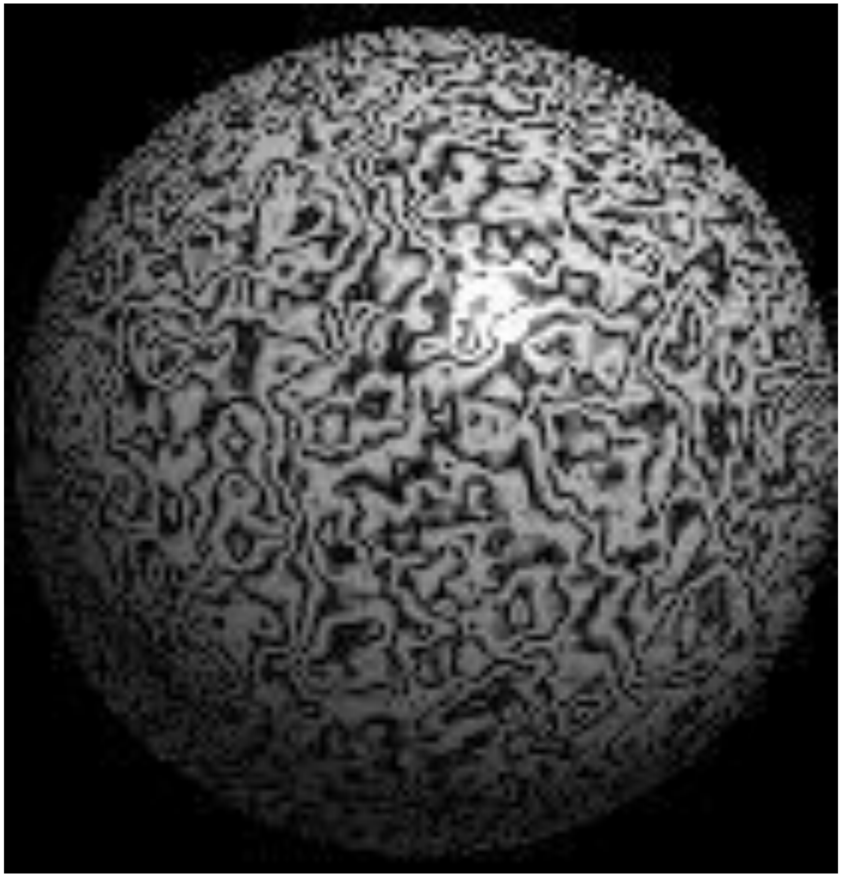
```
factorG  = sqrt(abs(sin(x + twist*turbulence(x, y, noise))))
color = (0, trunc(factorG*255), 255);
```

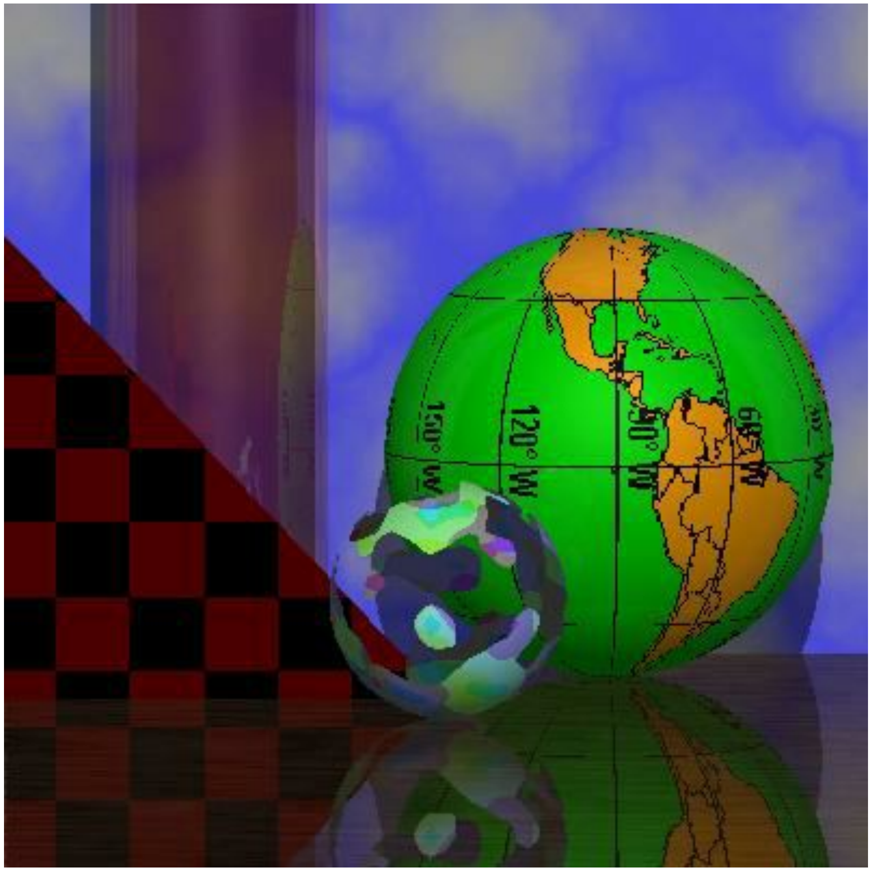
Clouds



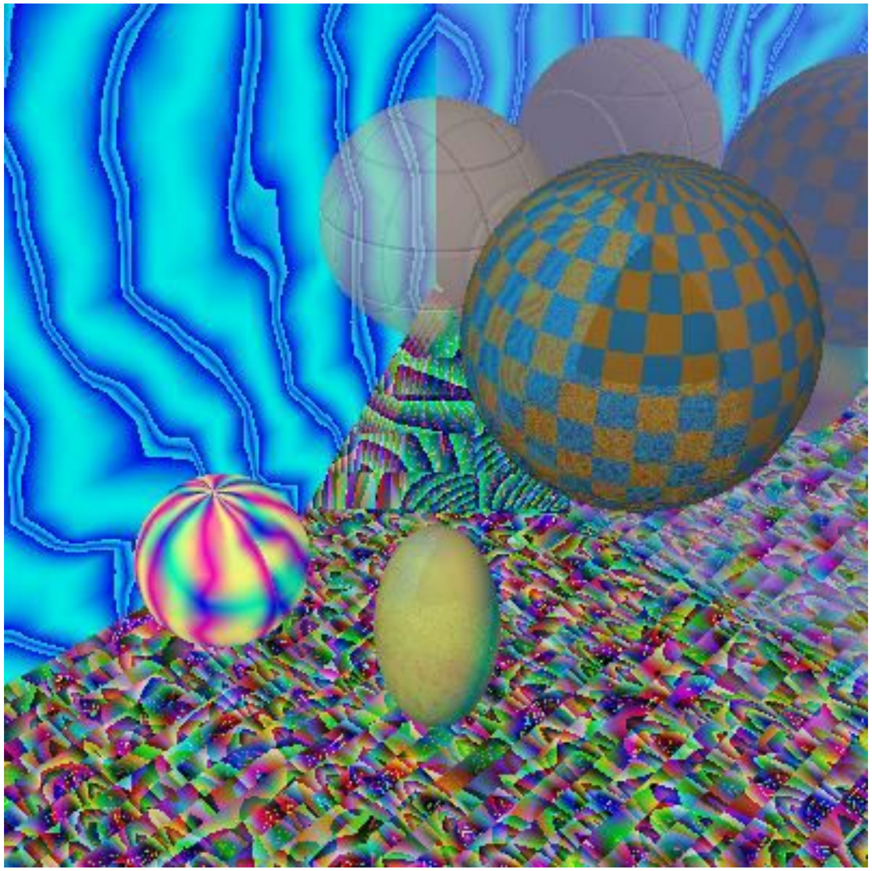
```
r = sqrt((x-200/d)*(x-200/d) + (y-200/d)*(y-200/d));
factorB = abs(cos(r + fluff*turbulence(x, y, noise));
color=(127 + 128*(1 - factorB), 127 + 128*(1 - factorB), 255);
```

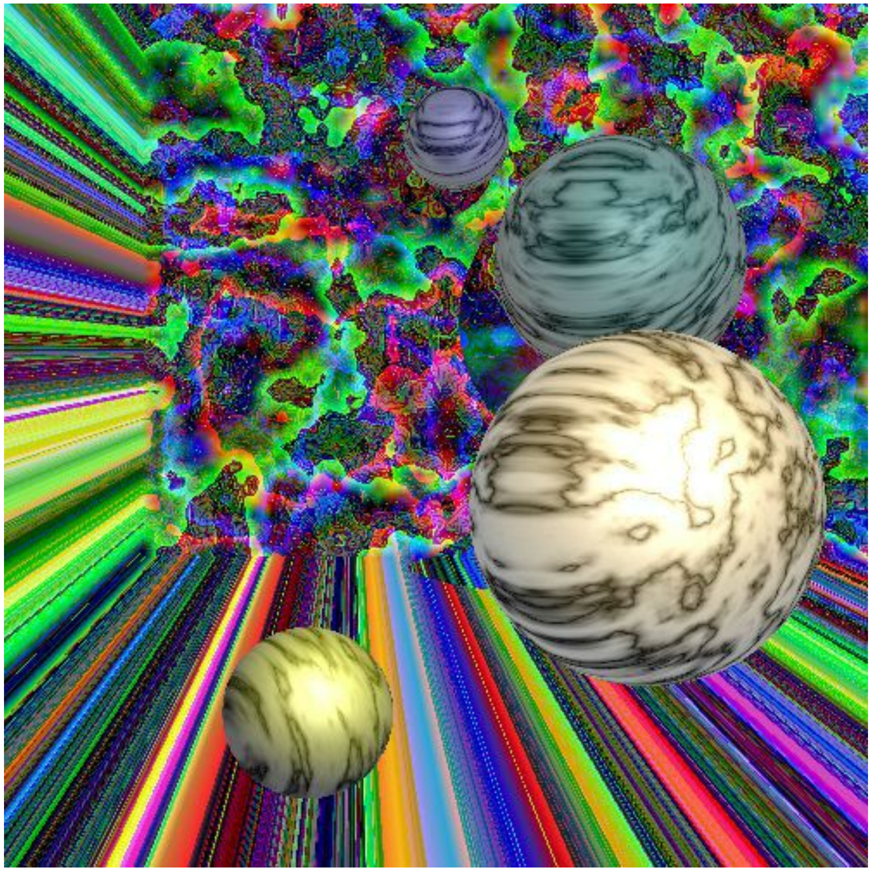
Student Images



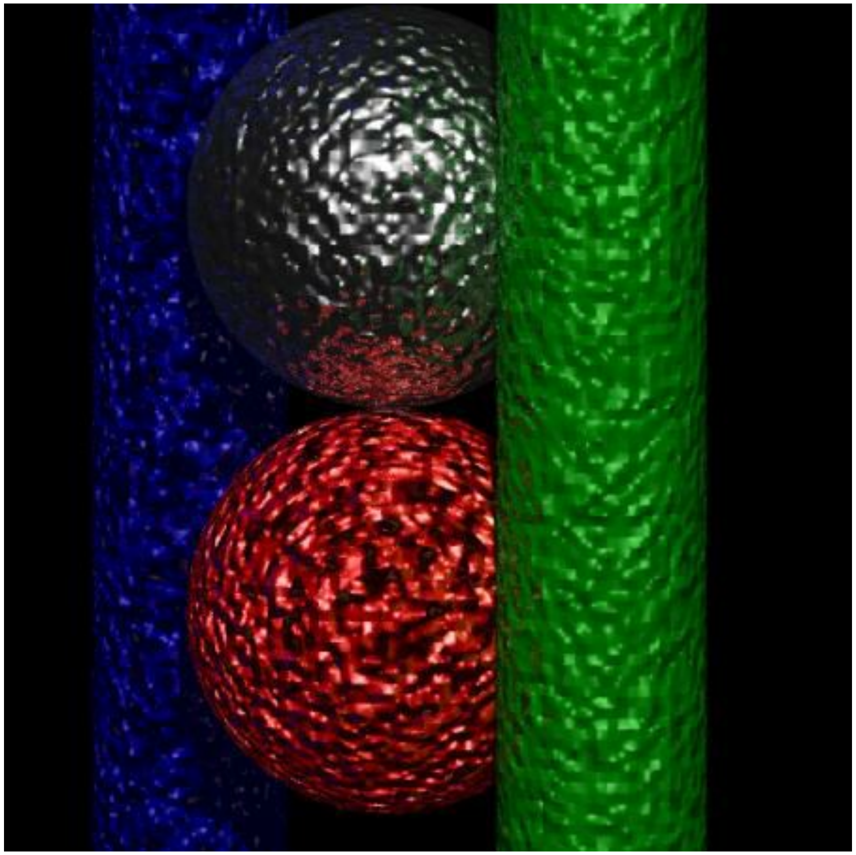


Student Images





Student Images



Perlin's Clouds and Corona