An Image Synthesizer

Ken Perlin (1985)
Presented by Marc Olano
Traditional Graphics

- Fixed functions
- Hard to change
- Simple
Pixel Stream Editor

- High-level programming
- Realistic stochastic natural texture
- Solid texture
Related Work

- Programmable Shading
  - Shade Trees [Cook 1984]

- Stochastic Texture
  - [Schacter 1980]
  - [Fournier et al. 1982]
  - [Gardener 1984]
Organization

- Introduction & Related Work
- **Pixel Stream Editor**
- Noise
- Conclusions
Pixel Stream Editor

- High-level language
- Runs on every pixel
  - Fat pixels [surface point normal …]
- Interpreted
- Fast design cycle
  - Edit + view low resolution < 1 minute
Includes *if*

```python
if surface == 1
    color = [1 0 0] *
    max(0.1, dot(normal, [1 0 0]))
else
    color = [0 0 0.1]
```

and loops

```python
f=1
while f < pixel_freq
    normal += Dnoise(f * point)
    f*=2
```
Language Features

- Indentation = nesting
- Scalar & vector variables
- User-defined functions
- Rich built-in functions
  - dot, norm, direction, Noise, …
Organization

- Introduction & Related Work
- Pixel Stream Editor
- Noise
- Conclusions
Noise

- Tool for Stochastic Solid Textures
  - Statistical invariance under rotation
  - Statistical invariance under translation
  - Narrow bandpass
Noise details

- When $x,y,z$ integer = integer lattice
- On lattice
  - Noise=0
  - Random gradient
    - Hash($x,y,z$)
- Off-lattice
  - Smooth interpolation
Compute Using Noise

Colorful(Noise(k * point))
Bumps & Noise Derivative

\[ \text{normal } \ += \ D\text{noise}(\text{point}) \]
Frequency Composition

f=1
while f < pixel_freq
    normal + = Dnoise(f * point)
f*=2
Turbulence

t = 0
scale = 1
while (scale > pixelsize)
    t += \text{abs}(\text{Noise}(p / scale) \times scale)
scale /= 2
return t
Turbulence & Diffraction
Organization

- Introduction & Related Work
- Pixel Stream Editor
- Noise
- Conclusions
Conclusions

- New approach for designing texture
- Fast and easy iterative design
- Powerful new Noise primitive
- Stochastic solid textures