Lesson 6 – Procedural Textures

PV227 – GPU Rendering

Jiří Chmelík, Jan Čejka
Fakulta informatiky Masarykovy univerzity

3. 11. 2015

Textures

- procedural textures,
  - stripes,
  - bricks,
  - “random” (fractal).

Texture Coordinates

- usually model specific,
- red $\rightarrow$ .s,
- green $\rightarrow$ .t.

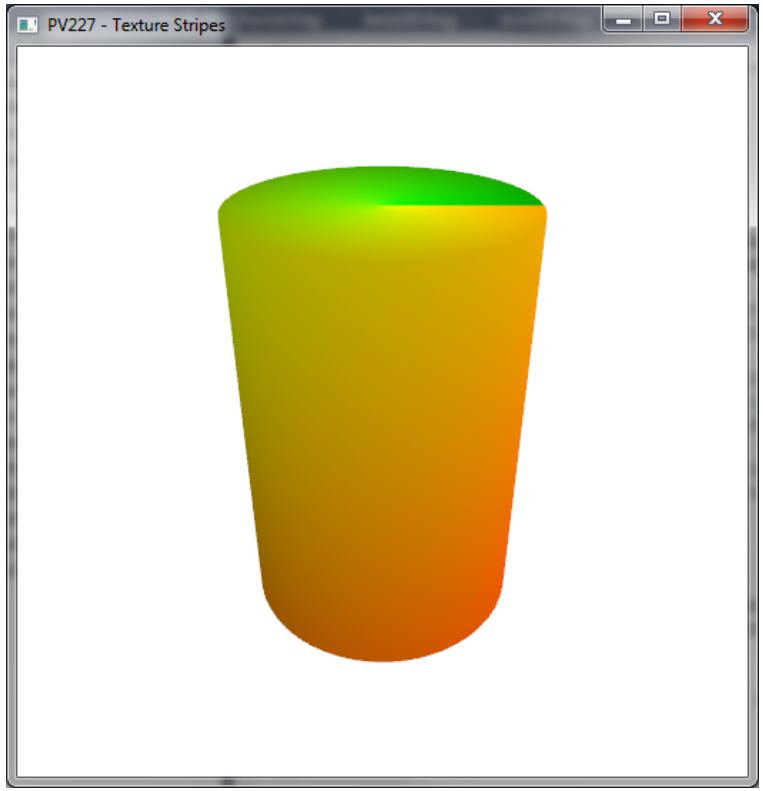


Figure: Visualization of texture coordinates

Texture Stripes

- interleave two colors in regular pattern,
- divide the [0,1] s-coordinate into multiple [0,1] ranges.

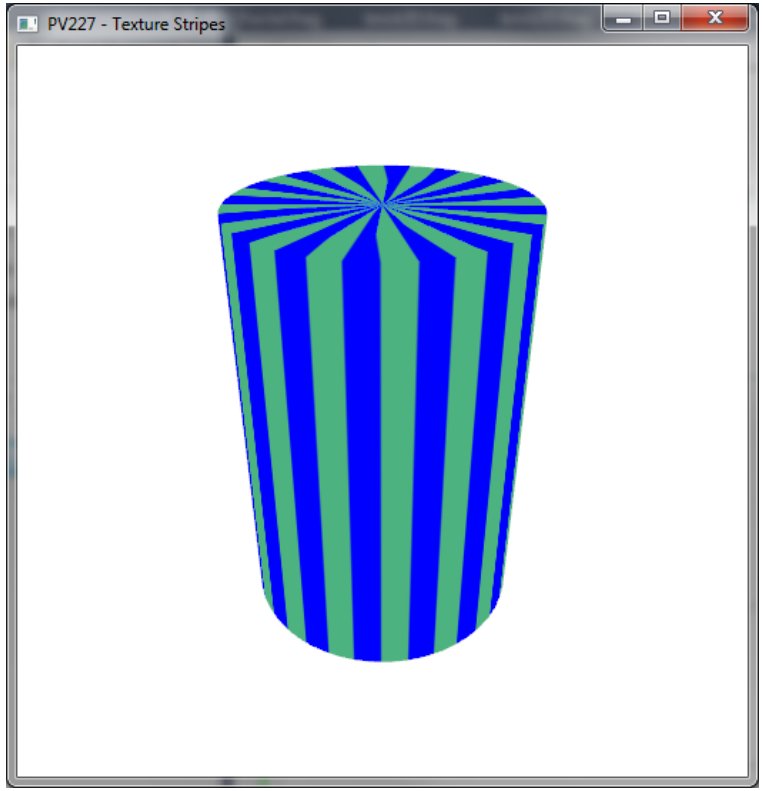


Figure: Stripe pattern

Texture Stripes – Color Mixing

- mix the two colors based on the position inside range,
- smooth the transition.

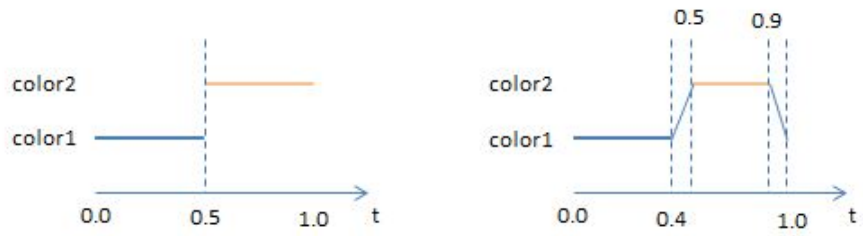
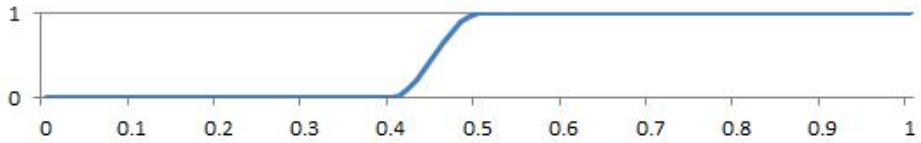


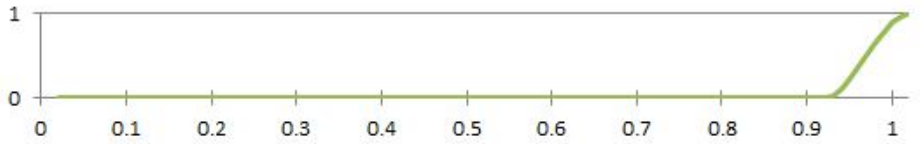
Figure: Taken from lighthouse3d.com

Texture Stripes – Smooth Interpolation

- f = smoothstep(0.4f, 0.5f, x);



- f = smoothstep(0.9f, 1.0f, x);



- f = smoothstep(0.4f, 0.5f, x) − smoothstep(0.9f, 1.0f, x);

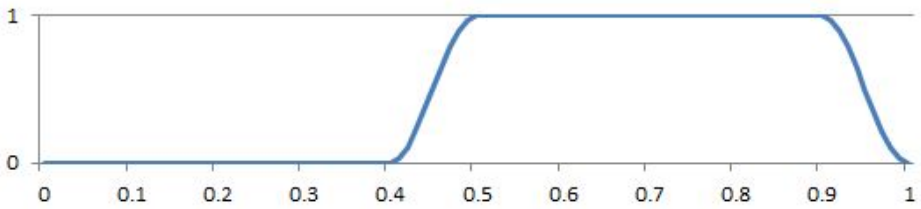


Figure: Taken from lighthouse3d.com

Brick 2D

- generating brick pattern in 2D,
- local space position or texture coordinate.

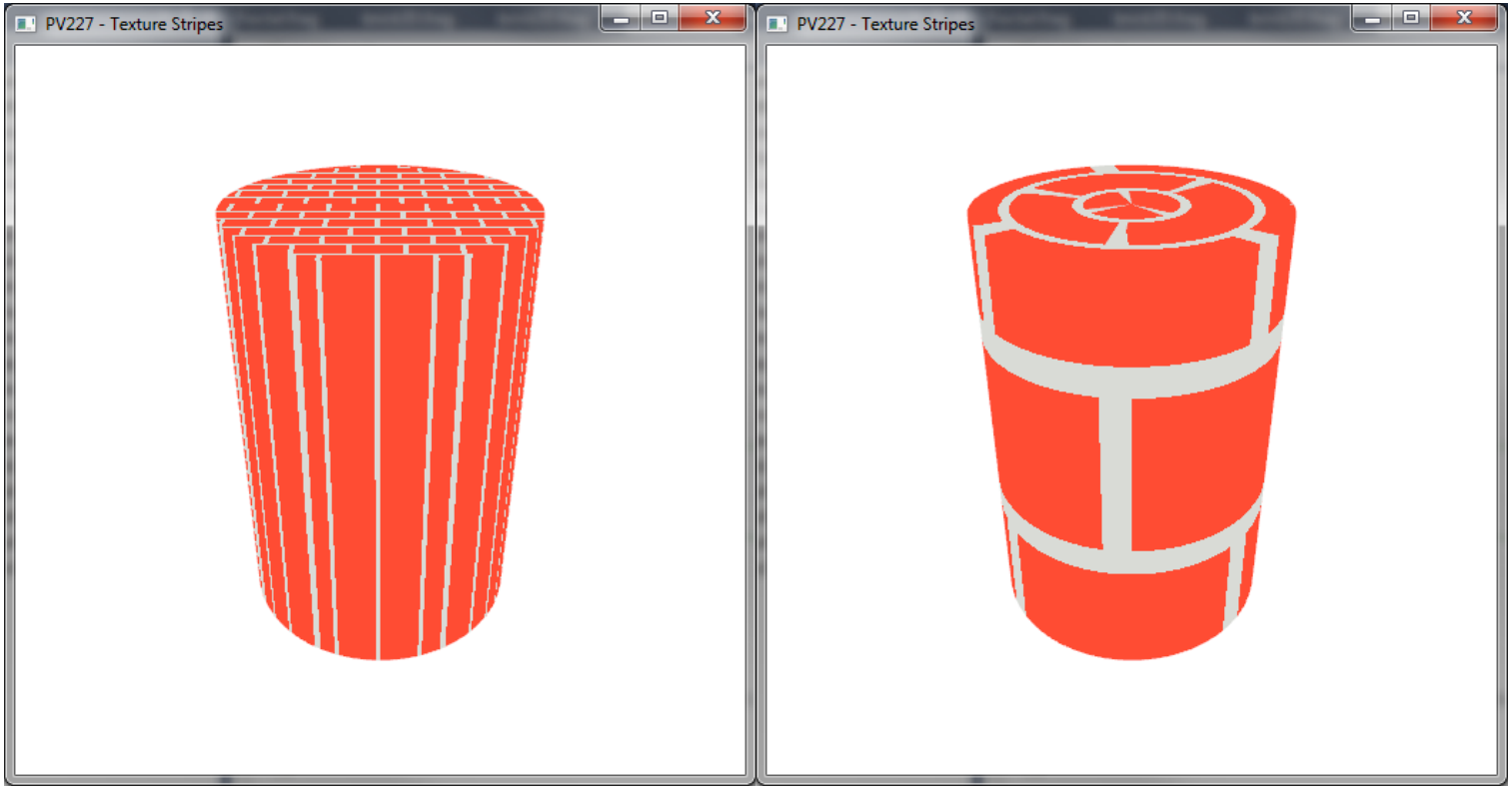


Figure: Brick pattern in 2D

Brick 2D – Schematics

- uniforms define the brick pattern,
- choose between the colors based on position relative to **BrickPct**.

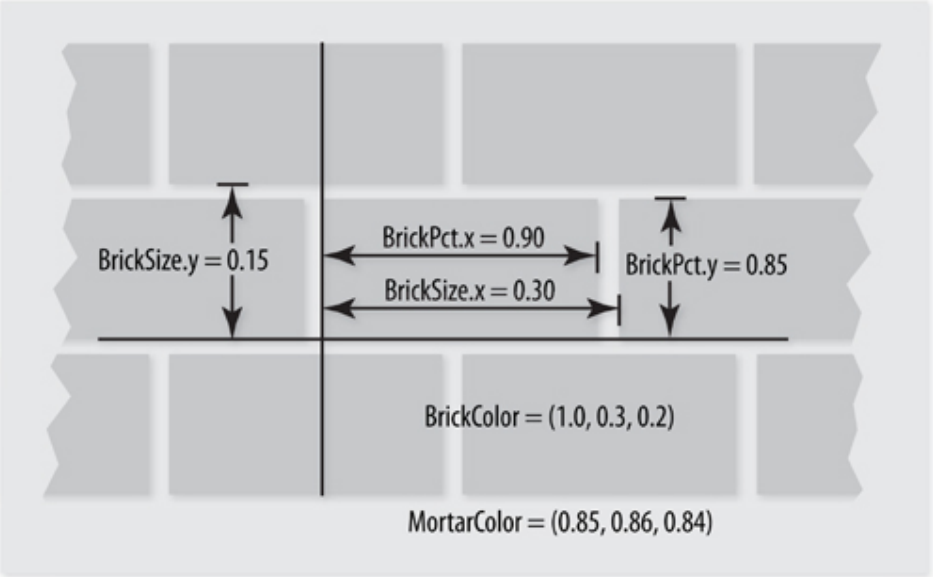


Figure: Taken from pearsoncmg.com

Brick 2D – Offset

- transform 3D space coordinates into 2D brick coordinates,
- compute the zigzag brick offset.

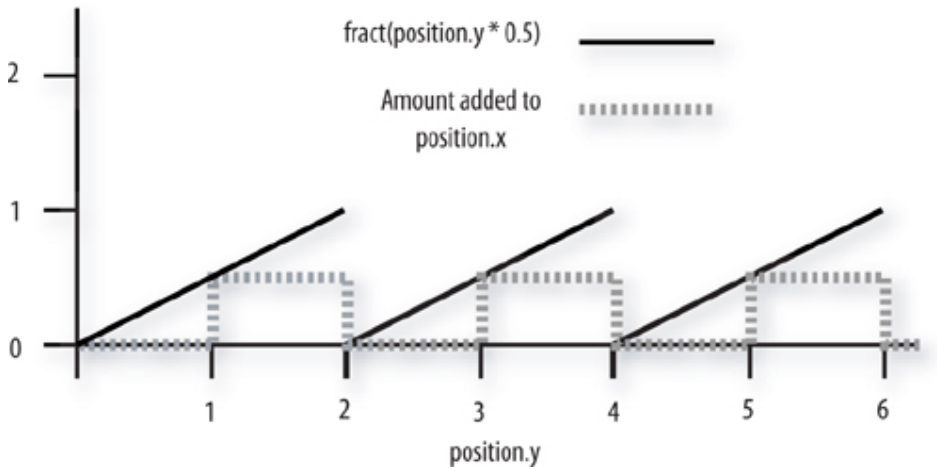


Figure: Taken from pearsoncmg.com

Brick 3D

- generate brick pattern in 3D,
- local space position or texture coordinate.

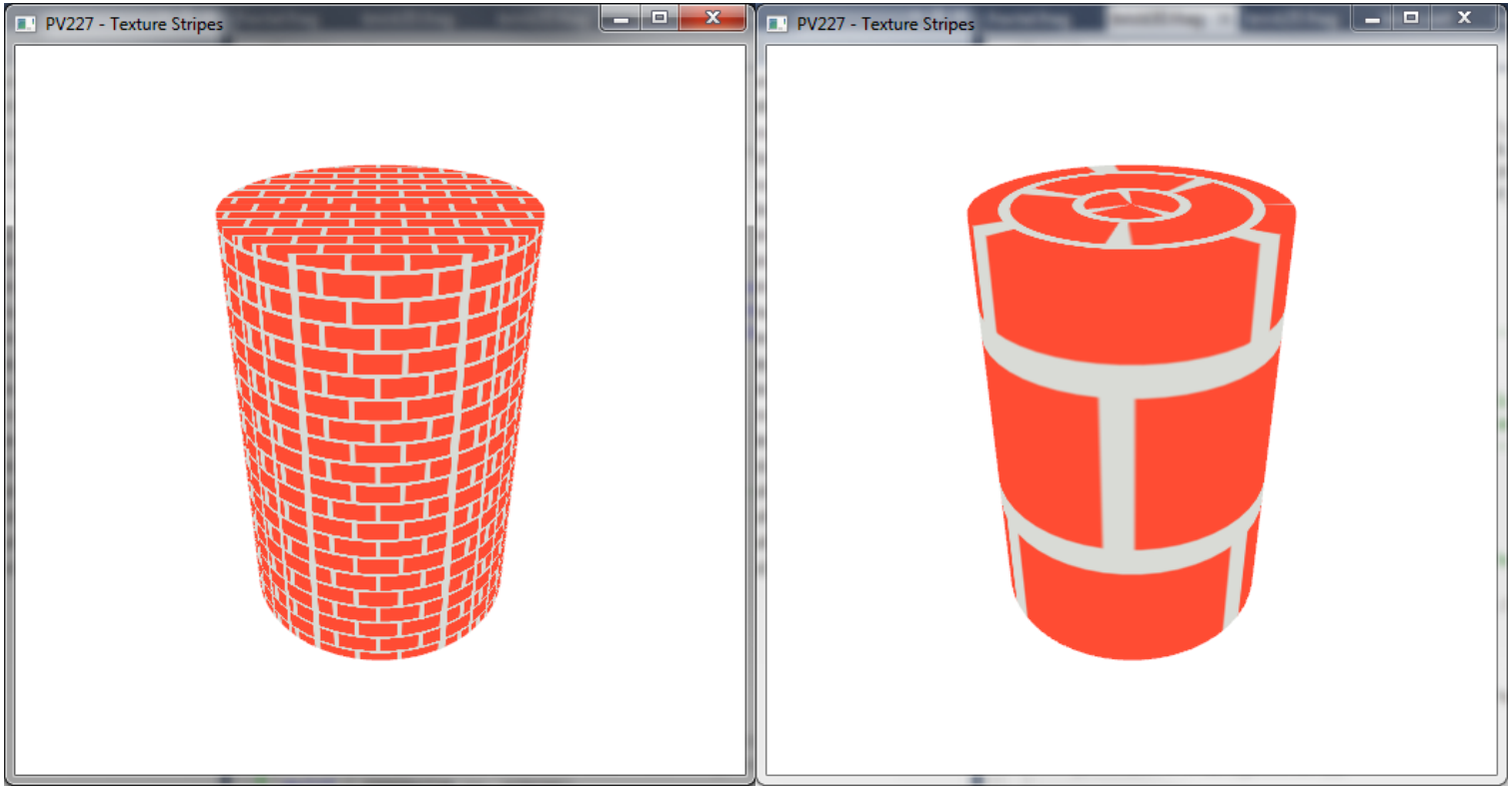


Figure: Brick pattern in 3D

Brick 3D – Offset

- same algorithm,
- zigzag brick offset needs logical XOR: `A != B`.

Fractals

- repeating the same pattern over and over,
- often starts from random values.

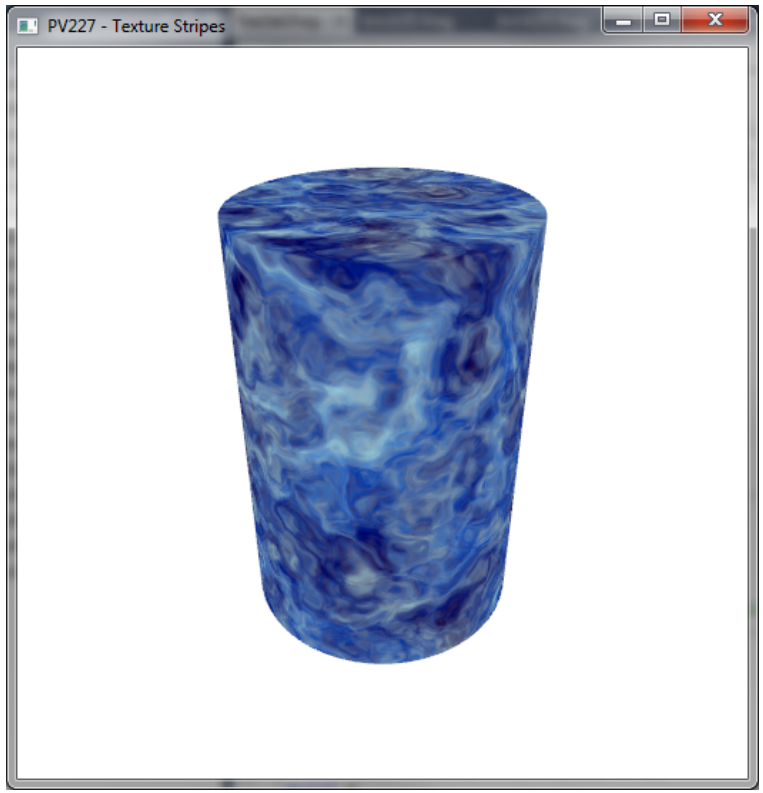


Figure: Fractal

Fractional Brownian Motion

- sum of a repeated pattern,
- half the amplitude, twice the frequency.

```
1  float fbm(vec3 p)
2  {
3    float f = 0.f;
4    f += 0.5000f * cnoise(p); p *= 2.0f;
5    f += 0.2500f * cnoise(p); p *= 2.0f;
6    f += 0.1250f * cnoise(p); p *= 2.0f;
7    f += 0.0625f * cnoise(p); p *= 2.0f;
8    f /= 0.9375f;
9
10   return f;
11 }
```

Fractional Brownian Motion – Iterations

- `fp = vec3(fbm(p));`
- `ffp = vec3(fbm(p + fp));`
- `fffp = vec3(fbm(p + ffp));`

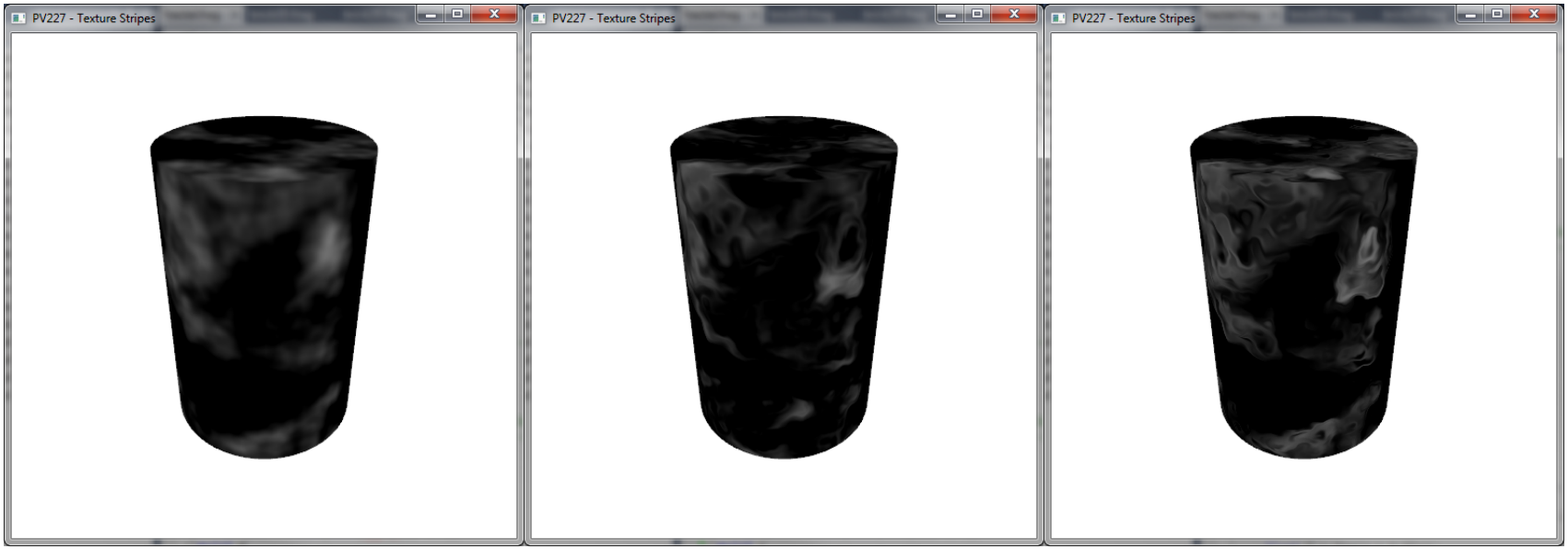


Figure: Increasingly detailed pattern

Good-looking Fractals

- combination of fixed constants and fbms,
- coefficients for mixing colors,
- look into the source code.