Procedural Solid Texturing

Project Report

02562 Rendering - Introduction
December 2025

Anton Thestrup Jorgensen
s194268

DANMARKS TEKNISKE UNIVERSITET
DTU COMPUTE

Contents

1 Introduction

Procedural textures generate surface appearance analytically, e.g. as a function of position, rather than relying on stored image data. When applicable, this approach offers several advantages, including the absence of texture memory usage, resolution independence, and consistent appearance under magnification. Because procedural textures are defined in continuous space, they integrate naturally with physically based rendering and path tracing pipelines.

Procedural textures are particularly well suited for materials with repetitive or stochastic structure, such as wood, marble, stone, clouds, smoke, or terrain, where the visual pattern follows geometric rules rather than specific imagery. They are also advantageous when materials must scale to large scenes, support extreme zoom levels, or remain stable under deformation and animation.

However, procedural textures are not universally appropriate. Materials that require precise, artist-authored detail (such as signage, text, logos, decals, or unique surface markings) are typically better represented using image-based textures. In these cases, direct control over exact pixel content outweighs the benefits of analytic generation. As a result, procedural and image-based textures are best viewed as complementary tools, each suited to different classes of materials and visual requirements.

2 Method

This project implements two procedural materials: a wood material and a marble material. Both are defined analytically in 3D space and evaluated at shading time, avoiding texture lookups. The materials are based on geometric projections, trigonometric modulation, and multi-octave noise.

Procedural Wood

The wood material models a cylindrical tree trunk with growth rings and longitudinal grain.

Log* Coordinate System (*the tree part, not the function) A central axis is defined by a point $\mathbf{p}_0$ and a unit direction $\mathbf{d}$. For a surface position $\mathbf{p}$, the coordinate along the trunk is

$$t = (\mathbf{p} - \mathbf{p}_0) \cdot \mathbf{d}.$$

The closest point on the axis is $\mathbf{q} = \mathbf{p}_0 + t\mathbf{d}$, and the radial vector in the plane orthogonal to the trunk is

$$\mathbf{r} = \mathbf{p} - \mathbf{q}, \qquad r = \|\mathbf{r}\|.$$

An orthonormal basis $(\mathbf{u}, \mathbf{v})$ spanning the plane orthogonal to $\mathbf{d}$ is constructed, allowing an angular coordinate around the trunk:

$$\theta = \operatorname{atan2}(\mathbf{r} \cdot \mathbf{v},\ \mathbf{r} \cdot \mathbf{u}).$$

Growth Rings Growth rings are modeled as a periodic function of the radial distance. To reproduce wider rings near the center and tighter rings near the bark, a nonlinear phase function is used:

$$\phi(r) = f\left(r + \alpha r^2\right),$$

where f is a base ring frequency and α controls the tapering of ring width. The raw ring signal is then

$$R(r) = \frac{1}{2}\left[1 + \sin(\phi(r))\right].$$

A smooth thresholding operation is applied to sharpen the contrast between earlywood and latewood regions.

Domain Warping To avoid perfectly concentric rings, the radial coordinate is perturbed using low-frequency noise:

$$\tilde{r} = r + \beta_1 N_1(\mathbf{p}) + \beta_2 N_2(\mathbf{p}),$$

where N_1 and N_2 are noise functions at different scales and β_i control the warp amplitude. This technique, known as domain warping, produces natural irregularities in the ring structure.

Color Composition Two base colors are defined for earlywood and latewood. The final diffuse color is computed by mixing these colors according to the ring signal and modulating the result by the grain term and an additional low-frequency tint variation.

Procedural Marble

The marble material is based on sinusoidal bands distorted by turbulent noise.

Vein Direction A unit vector $\mathbf{d}$ defines the dominant vein direction. For a point $\mathbf{p}$, a coordinate along this direction is

$$x = (\mathbf{p} - \mathbf{p}_0) \cdot \mathbf{d}.$$

Turbulence and Bands A multi-octave noise function $T(\mathbf{p})$ is evaluated and remapped to $[-1, 1]$ to produce turbulence. The marble pattern is generated by modulating a sinusoid with this turbulence:

$$M(\mathbf{p}) = \frac{1}{2}\left[1 + \sin\left(\omega x + \gamma T(\mathbf{p})\right)\right],$$

where ω controls band spacing and γ controls vein distortion. A smooth threshold is applied to emphasize veins.

Color Mapping Two colors are defined for the base stone and the veins. The final diffuse color is obtained by interpolating between them using the processed band signal.

Noise Functions

Both materials rely on procedural noise for natural variation.

Value Noise A scalar value noise function is defined on a 3D grid. For a point $\mathbf{p}$, the surrounding lattice points are assigned pseudo-random values via a hash function. Trilinear interpolation with a smoothstep kernel produces a continuous noise field:

$$N(\mathbf{p}) = \mathrm{lerp}_{xyz}\left(h(\lfloor \mathbf{p} \rfloor + \mathbf{i})\right),$$

where h is a hash function and $\mathbf{i} \in \{0, 1\}^3$.

Fractal Brownian Motion To obtain richer structure, multiple octaves of noise are summed:

$$\mathrm{fBm}(\mathbf{p}) = \sum_{k=0}^{K-1} a_k \, N(2^k \mathbf{p}),$$

with amplitudes a_k decreasing geometrically. This produces scale-invariant, natural-looking variation used for both domain warping and fine detail.

3 Implementation

This section explains the concrete WGSL implementation of the procedural noise functions and the two materials. My implementation is based on, and adds to, the outcome of Worksheet 7.

Noise Functions

All procedural variation is built on a compact value-noise implementation and its multi-octave extension.

Hash Function

Listing 1 shows a simple hash that maps a 3D point to a pseudo-random scalar in $[0, 1)$. It uses a dot product with large constants followed by a sine and fractional extraction.

Listing 1: Hash function

```
fn hash3(p: vec3f) -> f32 {
    let h = dot(p, vec3f(127.1, 311.7, 74.7));
    return fract(sin(h) * 43758.5453);
}
```

This function is inexpensive and sufficient for procedural textures.

Value Noise

Value noise is implemented by hashing the eight corners of the unit cube surrounding the query point and trilinearly interpolating between them. The implementation is shown in Listing 2.

Listing 2: 3D value noise

```
fn value_noise(p: vec3f) -> f32 {
    let i = floor(p);
    let f = fract(p);

    let u = f * f * (3.0 - 2.0 * f);

    let n000 = hash3(i + vec3f(0.0, 0.0, 0.0));
    let n100 = hash3(i + vec3f(1.0, 0.0, 0.0));
    let n010 = hash3(i + vec3f(0.0, 1.0, 0.0));
    let n110 = hash3(i + vec3f(1.0, 1.0, 0.0));
    let n001 = hash3(i + vec3f(0.0, 0.0, 1.0));
    let n101 = hash3(i + vec3f(1.0, 0.0, 1.0));
```

```
    let n011 = hash3(i + vec3f(0.0, 1.0, 1.0));
    let n111 = hash3(i + vec3f(1.0, 1.0, 1.0));

    let nx00 = mix(n000, n100, u.x);
    let nx10 = mix(n010, n110, u.x);
    let nx01 = mix(n001, n101, u.x);
    let nx11 = mix(n011, n111, u.x);

    let nxy0 = mix(nx00, nx10, u.y);
    let nxy1 = mix(nx01, nx11, u.y);

    return mix(nxy0, nxy1, u.z);
}
```

The cubic polynomial used to compute u is a smoothstep function, ensuring continuous first derivatives across cell boundaries.

Fractal Brownian Motion

Fractal Brownian motion (fBm) is implemented as a fixed sum of five octaves of value noise (Listing 3). Frequency doubles and amplitude halves each octave.

Listing 3: Fractal Brownian motion

```
fn fbm(p: vec3f) -> f32 {
    var sum = 0.0;
    var amp = 0.5;
    var freq = 1.0;
    for (var i = 0; i < 5; i++) {
        sum += amp * value_noise(p * freq);
        freq *= 2.0;
        amp *= 0.5;
    }
    return sum;
}
```

This produces smooth, scale-invariant noise used for both domain warping and fine detail.

Procedural Wood Material

The wood material implementation is shown in Listing 4. It models a tree trunk using an axial coordinate system, growth rings, and longitudinal grain.

Listing 4: Procedural wood material

```
fn progressive_material_wood(pos: vec3f) -> Material {
    let center_pos = vec3f(25.0, 0.0, 25.0);
    let d = normalize(vec3f(0.2, 1, -0.1));

    let up = select(
```

```
        vec3f(0.0, 1.0, 0.0),
        vec3f(1.0, 0.0, 0.0),
        abs(d.y) > 0.95
    );
    let u = normalize(cross(up, d));
    let v = cross(d, u);

    let pd = pos - center_pos;
    let t = dot(pd, d);
    let proj = center_pos + t * d;
    let radial = pos - proj;
    let r = length(radial);

    let p = pos * 0.20;
    let turb = 2.0 * fbm(p * 2.0) - 1.0;
    let turb2 = 2.0 * fbm(p * 6.0 + 13.7) - 1.0;
    let warp_r = r + 0.35 * turb + 0.10 * turb2;

    let ring_freq = 0.5;
    let taper = 0.25;

    let phase = ring_freq * (warp_r + taper * warp_r * warp_r);

    let rings = 0.5 + 0.5 * sin(phase);
    let ring_sharp = smoothstep(0.25, 0.85, rings);

    let angle = atan2(dot(radial, v), dot(radial, u));
    let grain = fbm(vec3f(t * 0.6, angle, 0.0) + pos * 0.05);
    let grain_contrast = smoothstep(0.35, 0.75, grain);


    let dark_wood  = vec3f(0.33, 0.17, 0.08);
    let light_wood = vec3f(0.62, 0.42, 0.22);

    var color = mix(dark_wood, light_wood, ring_sharp);

    color *= mix(0.85, 1.05, grain_contrast);

    let tint = fbm(pos * 0.08 + 4.2);
    color *= (0.95 + 0.10 * tint);

    return Material(
        vec3f(0.0),
        color
    );
}
```

Coordinate System The trunk axis is defined by `center_pos` and direction `d`. An orthonormal basis $(\mathbf{u}, \mathbf{v}, \mathbf{d})$ is constructed using cross products. A conditional choice of the reference "up" vector avoids numerical instability when `d` is nearly vertical.

Each shading point is decomposed into:

$$t = (\mathbf{p} - \mathbf{p}_0) \cdot \mathbf{d}, \qquad r = \|\mathbf{p} - (\mathbf{p}_0 + t\mathbf{d})\|.$$

Domain Warping The radial distance r is perturbed using two fBm evaluations at different frequencies. This step (lines computing `turb`, `turb2`, and `warp_r`) introduces irregular, natural-looking ring deformation.

Growth Rings Ring spacing is controlled by a nonlinear phase:

$$\phi(r) = f\,(r + \alpha r^2),$$

implemented via `ring_freq` and `taper`. This produces wide inner rings and progressively tighter outer rings. The sinusoidal signal is sharpened using `smoothstep` to emphasize contrast between earlywood and latewood.

Longitudinal Grain The angular coordinate

$$\theta = \operatorname{atan2}(\mathbf{r} \cdot \mathbf{v}, \mathbf{r} \cdot \mathbf{u})$$

is combined with the axial coordinate t and evaluated through fBm. This creates streaks aligned with the trunk.

Color Composition Two fixed colors represent dark and light wood. Ring structure selects between them, while grain and a low-frequency tint variation further regulate the result. The material has no emission and is purely diffuse.

Procedural Marble Material

The marble material is shown in Listing 5.

Listing 5: Procedural marble material

```
fn progressive_material_marble(pos: vec3f) -> Material {
    let center_pos = vec3f(0.0, 0.0, 0.0);
    let d = normalize(vec3f(0.2, 0.6, 0.77));

    let x = dot(pos - center_pos, d);

    let p = pos * 0.15;
    let t = fbm(p * 2.0);
    let warp = 2.0 * t - 1.0;

    let frequency = 3.0;
    let phase = x * frequency + 10.0 * warp;
    let bands = 0.5 + 0.5 * sin(phase);
```

```
    let veins = smoothstep(0.25, 0.75, bands);

    let base = vec3f(0.92, 0.92, 0.94);
    let vein = vec3f(0.20, 0.20, 0.25);

    let color = mix(vein, base, veins);

    return Material(
        vec3f(0.0),
        color
    );
}
```

Directional Parameterization A dominant vein direction `d` defines a scalar coordinate

$$x = (\mathbf{p} - \mathbf{p}_0) \cdot \mathbf{d}.$$

Turbulence and Veins An fBm-based turbulence term perturbs the phase of a sinusoidal function of x:

$$M(\mathbf{p}) = \frac{1}{2}\left[1 + \sin(\omega x + \gamma T(\mathbf{p}))\right].$$

This produces characteristic marble veins. A smoothstep operation sharpens the result.

Color Mapping Two colors define the base stone and the veins. The final diffuse color is obtained by interpolating between them using the processed band signal.

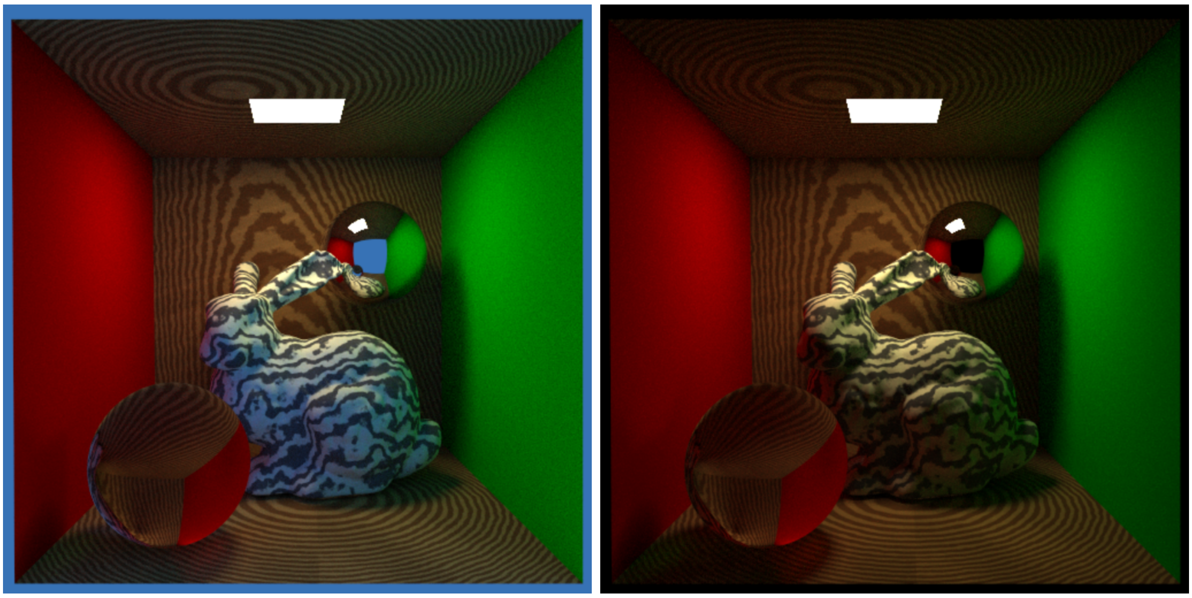

Figure 1: Final render of the scene with procedural wood and marble materials using different lighting conditions.

4 Results

Figure 1 shows the final scene; the Cornelle Box (no blocks), populated by the Stanford Bunny and two spheres using a mirror and refractive shader respectively. It is rendered with two procedural solid materials: wood on the previously gray surfaces of the Cornell Box, and marble on the Stanford Bunny.

The wood material produces visible growth rings with irregular ring boundaries and a weaker longitudinal grain component. The ring structure is wider near the center of the log, and tapers off with distance. The noise is perhaps a bit too aggressive, and a stronger angular dependency could be implemented to make the rings less circular if desired.

The marble material produces banded veining aligned with a single dominant direction. The veining thickness and spacing vary smoothly due to turbulence, and the transition between vein and base stone is fairly sharp. The noise level works well for the 3D structure/scale of the bunny, and produces both bands and specks.

The two images use different lighting conditions via a blue and black background respectively. Under cooler illumination the marble appears higher-contrast and the wood reads slightly more desaturated, while warmer illumination emphasizes the latewood/earlywood color difference and reduces the perceived contrast of the marble veins. The lower overall lighting in the black background case also contributes to this effect.

5 Discussion

Wood. The wood appearance comes primarily from a radial phase function (rings) combined with domain warping and an fBm-based grain term. Domain warping is important: without it, the rings would be perfectly concentric and the material would look synthetic. With warping, ring spacing and ring boundaries become locally irregular, which better matches real growth variation.

A limitation is that the current model is purely diffuse. Under strong illumination the wood lacks view-dependent effects such as a subtle clearcoat or specular sheen, which are common for varnished or polished wood. Adding a specular lobe (and optionally a clearcoat layer) would improve realism while keeping the same diffuse albedo model. For non-treated wood, an improvement could be in reproducing splinters and bumps with a procedural normal map.

Marble. The marble model uses a sinusoidal band function whose phase is perturbed by turbulence. This reliably produces veins and keeps them continuous in 3D. The smooth thresholding step increases vein contrast, which helps readability at typical viewing distances, but it also risks making the veins look too binary if the threshold is too aggressive. A softer mapping (or a multi-band color ramp) could preserve contrast while keeping more mid-tones.

Interaction with lighting. Because both materials are mostly low-frequency albedo modulation (diffuse), the perceived quality depends strongly on illumination spectrum and intensity. In the cooler lighting, the marble's dark veins separate more strongly from the base, while warmer lighting visually compresses that contrast. This is expected: the models encode structure in albedo, so color temperature shifts and global exposure changes directly affect apparent contrast and material identity.

Numerical and implementation considerations. The noise implementation uses hashed value noise with smooth interpolation, and fBm sums multiple octaves. This is efficient and stable for real-time shading, but value noise can show mild grid-aligned bias in some cases. If this becomes visible (e.g., faint axis-aligned banding on large smooth surfaces), switching to gradient noise or applying a small domain rotation per octave can reduce directional artifacts.

Overall. Two solid procedural materials with continuous 3D structure and no texture lookups—is achieved. The wood reads as ring-based with natural irregularity, and the marble reads as directionally veined with turbulent distortion. The most direct improvements would be (1) adding a specular/clearcoat component (or mesoscale geometry) for wood, (2) refining the marble color mapping beyond two colors, and (3) reducing potential grid artifacts by improving the noise basis.