Multi-Dimensional Procedural Wave Noise

Pascal Guehl[1], R. Allègre[2], G. Gilet[3], B. Sauvage[2], M-P. Cani[1], J-M. Dischler[2]

[1]LIX, Ecole Polytechnique, CNRS, IP Paris, France [2]ICube, Université de Strasbourg, CNRS, France
[3]Université de Sherbrooke, Canada

SIGGRAPH 2025
Vancouver+ 10-14 August

Problem

Procedural noise is widely used as a **texturing and modeling tool** [2,6] to add visual complexity in computer graphics.

However, achieving **spectral control** in **3D, animated noise (3D+T), and higher dimensions** becomes **computationally expensive** and **memory-intensive**, especially on GPUs.

Related Work / Motivation

Contrary to Perlin noise, the sparse convolution family [1] — including **Gabor** [3], **LRP** [4], and **Phasor** [5] — offers spectral control, but at a **high computational cost**.

However, noise methods often lack **key desired properties** such as **resolution-independence**, **infinite spatial extent**, **spectral control**, **GPU efficiency**, **low memory usage**, and **scalability to arbitrary dimensions** like **2D, 3D, 3D+time**.

Isotropic and anisotropic behavior is also not consistently supported.

Our Approach / Solution

We propose a new **procedural noise model** based on the **superposition of randomly oriented hyperplanar waves** with random phases.

This wave-based formulation efficiently reproduces existing noise types (e.g., **Gabor**, **Phasor**, **Gaussian**, and **ridged** noise), while preserving essential procedural properties such as **infinite extent**, **resolution independence**, and **GPU-friendly evaluation**.

Our method naturally scales to **3D, 3D+T**, and even higher dimensions, and supports both **isotropy** and **anisotropy**, **spectral control** — all with **minimal data** and **low memory usage**.

References

[1] Lewis. *Algorithms for Solid Noise Synthesis*. SIGGRAPH, 1989.

[2] Ebert et al. *Texturing and Modeling: A Procedural Approach*. Elsevier, 2002.

[3] Lagae et al. *Procedural Noise Using Sparse Gabor Convolution*. ACM TOG, 2009.

[4] Gilet et al. *Local Random-Phase Noise for Procedural Texturing*. ACM TOG, 33(6), 2014.

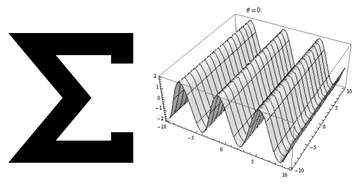
[5] Tricard et al. *Procedural Phasor Noise*. ACM TOG, 38(4), 2019.

[6] Guehl et al. *Semi-Procedural Textures Using Point Process Texture Basis Functions*. CG Forum, 39(4), 2020.

Overview of our Core Features

Wave-based Model

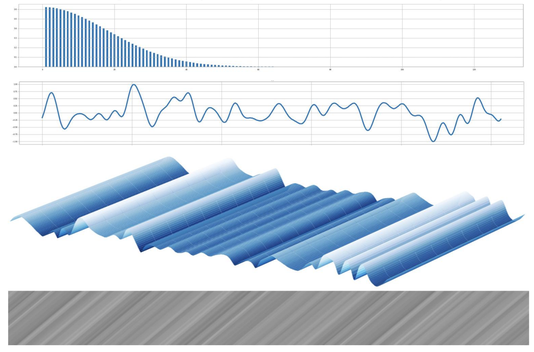
Sum of ***randomly oriented*** *(ξ)* **hyper-planar waves** with ***random phase*** *(ϕ)*.



$$A(\xi)e^{i(2\pi\,\xi\cdot\mathbf{x}-ct+\phi)}$$

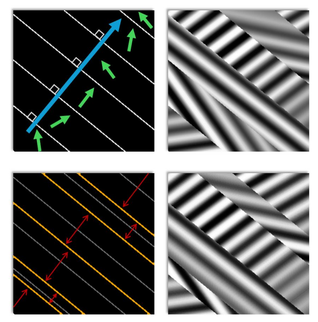
Spectral Control

Precomputed amplitudes *A(ξ) as* **1D profile** (compact data on GPU).



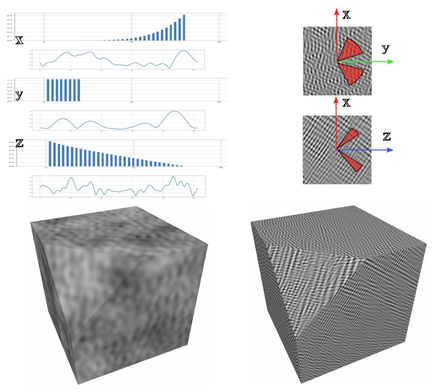
Random Tiling & Blending

Slicing: add variability (*jittering*), coarser direction sampling (*less computation*), prevent alignment artefacts, faster neighbors access.



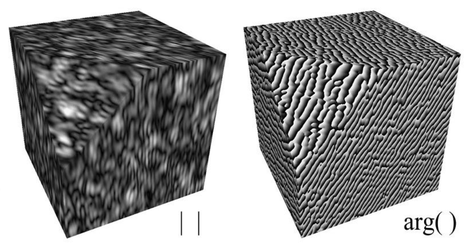
Anisotropy

Left: **different amplitudes** *A(ξ) along axes x,y,z*. **Right**: same amplitude *A(ξ) is null outside* ***different solid angles***.



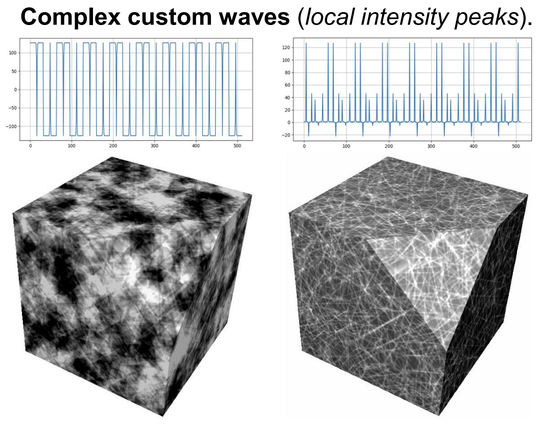
Phasor and Ridged noises

Solid wave noise is **complex valued**.



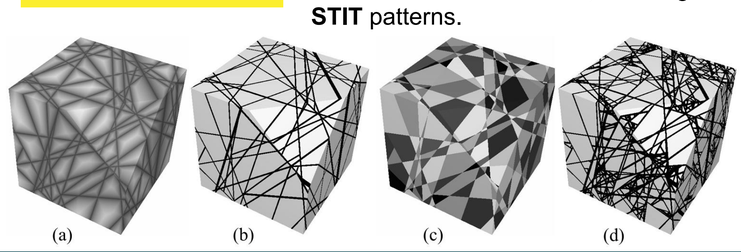
Crystal-like and Wired noises

Complex custom waves (*local intensity peaks*).



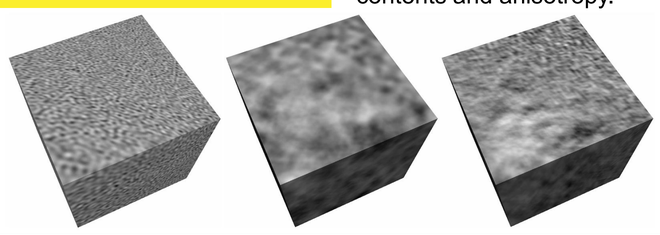
Cellular noise

Different from Worley noise, we use **iterative cell subdivision**, imitating **STIT** patterns.



Smooth Transitions

Varying frequency range, contents and anisotropy.

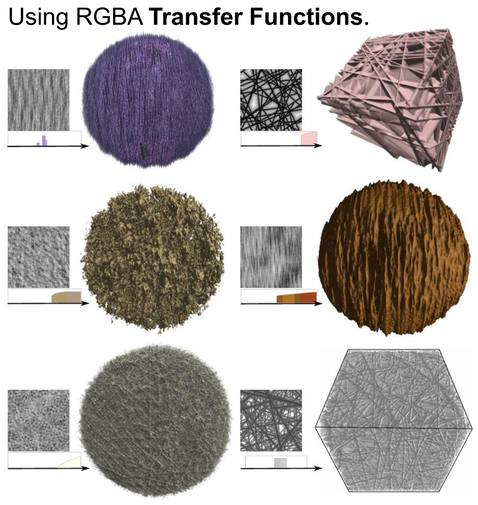


Results

(see our ***supplemental materials*** *for more results and comparisons)*

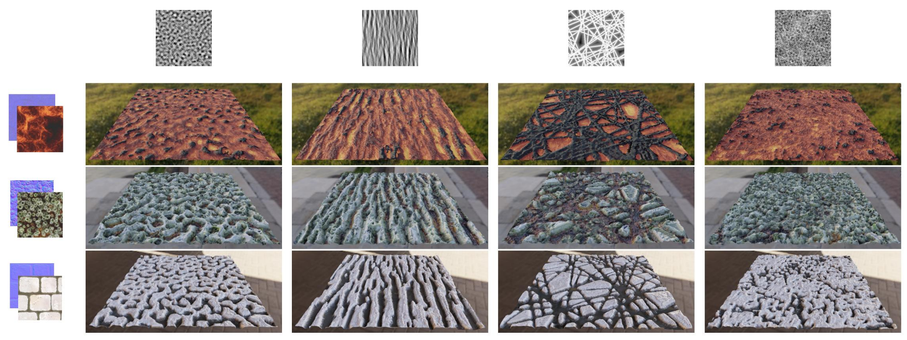
(by-example) Volumetric Materials

Using RGBA **Transfer Functions**.



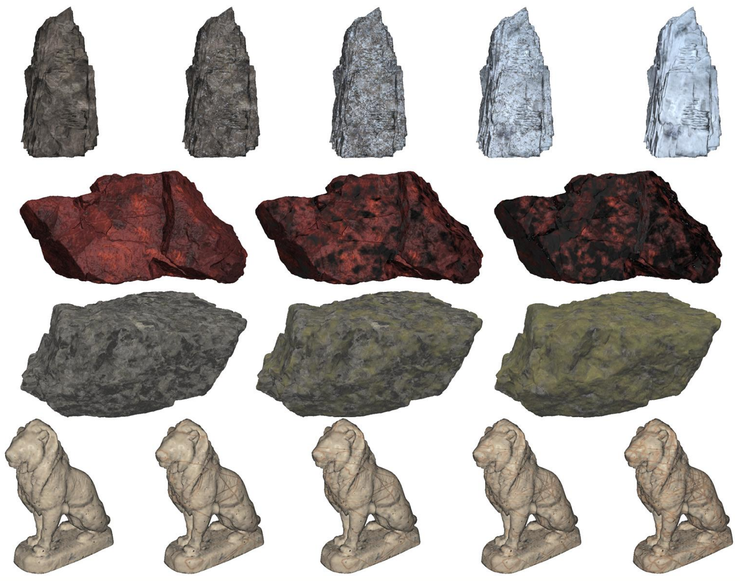
(by-example) PBR Materials

Using **Semi-Procedural Textures** [6] to guide *details* by our wave noise *structure*.



Animated Materials

3D+T wave noise for **material animation**: freezing rock, cooling a glowing stone, moss growth, cracks forming on a clay statue.



3D Mesh Texturing *(without UVs)*

Using **Style Transfer Functions** (left) from volume rendering.