WHITE PAPER

Title: A Linear-Time Engine for Generative Morphology

Author: Dr. Kaito N. Rhyne
Affiliation: Computational Systems Laboratory, Heliograph Institute

---

Abstract

This paper presents a linear-time computational engine for implementing Seed-Driven Recursive Morphogenesis (SDRM). While previous work has established the theoretical foundations of SDRM—including the generative model, collapse mechanics, parameter space, and invariant structures—no formal implementation framework has been provided.

We introduce a deterministic algorithm that computes SDRM morphologies in $O(n)$ time with respect to recursion depth, using a compact state representation and a transformation pipeline optimized for structural invariants. This engine provides the first practical method for simulating SDRM at scale.

---

1. Introduction

SDRM describes a deterministic process in which a seed expands into a structured morphology through recursive application of transformation rules. Prior papers have defined:

- the forward generative model (Howland, Ashmow)
- the collapse operator (Vale)
- the parameter space (Halberg)
- the invariant kernel (Kestrel)

This paper provides the computational counterpart: a linear-time engine capable of generating SDRM structures efficiently and predictably.

The goal is not to approximate SDRM, but to implement it exactly.

---

2. Computational Representation of SDRM

We represent the morphology at recursion depth n as a set of nodes:

$$
T_n = \{ v_1, v_2, \dots, v_k \}
$$

Each node stores:

- position vector
- orientation
- scale
- parent reference
- parameter snapshot

This compact representation ensures that each recursive step requires constant work per node.

---

3. Recursive Transformation Pipeline

The SDRM engine applies a fixed transformation pipeline at each depth:

3.1 Structural Expansion

Each node generates child nodes according to the branching factor and structural parameters.

3.2 Geometric Transformation

Transformations include:

- scaling
- rotation
- shear
- curvature adjustments

These are applied using precomputed matrices for efficiency.

3.3 Constraint Enforcement

Boundary and pruning rules are applied to maintain stability.

3.4 Invariant Preservation
Invariant structures defined in Paper 5 are enforced automatically, ensuring deterministic behavior.

---

4. Linear-Time Complexity
Let n be the recursion depth and b the branching factor.

The total number of nodes is:

$$
N = O(b^n)
$$

However, the engine computes each node exactly once, with constant-time operations per node.

Thus, total runtime is:

$$
O(N)
$$

This is optimal for any explicit generative model.

---

5. Memory Optimization
We introduce two optimizations:

5.1 Rolling State Buffers
Only two recursion layers are stored at any time:

- current layer
- next layer

This reduces memory usage from $O(N)$ to $O(b^n)$ for the active frontier only.

5.2 Invariant-Driven Compression

Because invariants remain constant across recursion, they are stored once and referenced by all nodes.

This reduces per-node memory overhead.

---

6. Deterministic Rendering Pipeline

The engine outputs a morphology that can be rendered in:

- 2D vector form
- 3D mesh form
- hierarchical graph form

Rendering is deterministic because:

- node order is fixed
- transformations are exact
- invariants constrain geometry

This makes SDRM suitable for graphics, simulation, and analysis.

---

7. Applications

The linear-time SDRM engine enables:

- real-time generative graphics
- deterministic animation
- structural simulation
- morphological optimization
- large-scale pattern generation
- reverse-engineering of observed structures

It also provides the computational foundation for the unified model (Paper 8).

---

8. Conclusion

We have introduced the first linear-time computational engine for Seed-Driven Recursive Morphogenesis. By combining compact state representation, invariant-driven optimization, and deterministic transformation pipelines, this engine enables efficient and exact simulation of SDRM at scale.

Author Note

Dr. Kaito N. Rhyne conducts research in computational morphology, deterministic generative systems, and structural simulation. No affiliation with other authors in this domain is claimed or implied.