Aesthetic DNA: A Calculus of Form, Flow, and Feeling

Version: 0.8 (Draft)
Date: November 6, 2025
Project: aeDNA — The Aesthetic Genome Lab
Branches: Calculus • Chemistry • Thermodynamics

Abstract

We propose **Aesthetic DNA (aeDNA)**: a practical, quantitative framework for analyzing and synthesizing visual form through (1) **Aesthetic Calculus** (multi-scale geometry and vector fields), (2) **Aesthetic Chemistry** (palette, materiality, and transformation kinetics), and (3) **Aesthetic Thermodynamics** (order–disorder, energy flows, and attractor structure).

The system ingests media (images initially), computes a compact set of interpretable field- and graph-level features, and fuses them into an **Aesthetic Complexity Index (ACI)** and a **Complexity Atlas** of diagnostic maps. This white paper formalizes metrics, gives a modular software spec, and sets a roadmap for research-grade validation and creative tooling.

---

1. Motivation

Artists, designers, and critics routinely speak about **movement, density, tension, breathing space, rhythm, gesture, and flow**. These are geometric and energetic intuitions. aeDNA translates these intuitions into measurable objects while **preserving interpretability**:

- **Field language** for gesture (gradients, orientations, vector curls/divergences).
- **Topology** for persistence of shapes across thresholds.
- **Graph structure** for skeletal anatomy of forms.
- **Information measures** for order vs. noise.
- **Palette kinetics** for chromatic chemistry and material metaphors.

The goal is not to reduce art to numbers; it is to provide **sharp instruments** that reveal structure, compare variations, and fuel creative iteration.

---

2. System Overview

Inputs: raster images (PNG/JPEG/WEBP). Future: video, 3D scans, vector drawings.

Pipelines (branches): - **Calculus:** gradient fields → orientation coherence, curl/divergence, skeleton graphs, persistence barcodes, fractal dimension (box-counting), lacunarity. - **Chemistry:** palette extraction, gamut area, hue–chroma–lightness distributions, complementary tension score, local palette diffusion ("chromatic reaction–diffusion" proxy), simulated pigment-mixing operators. - **Thermodynamics:** multi-scale entropy (Shannon/Levine), compressibility (PNG/WEBP dictionary size proxy for Kolmogorov $\hat{K}$), spatial temperature field (variance-at-scale), attractor sketches (iterated filtering trajectories), and **exhaustion curves**.

Outputs: per-branch metrics; ACI; **Complexity Atlas** (small-multiples of maps: coherence, curl, divergence, lacunarity, persistence span, palette fields, entropy fields).
Interfaces: React/Tailwind UI with exportable JSON/CSV, printable PDF, and side-by-side comparisons.

---

3. Aesthetic Calculus (Formalism)

Let $I : \Omega \subset \mathbb{R}^2 \to \mathbb{R}^3$ be an image; let L be a luminance channel, C a perceptual color embedding (e.g., OKLab). Compute gradient ∇L , structure tensor $J = G_\sigma * (\nabla L \, \nabla L^\top)$.

3.1 Orientation Coherence (Flow Order)

For each tile or pixel neighborhood, let $\lambda_1 \geq \lambda_2$ be eigenvalues of J . Define **coherence** $\kappa = (\lambda_1 - \lambda_2)/(\lambda_1 + \lambda_2 + \epsilon)$.
Maps: coherence heatmap and histogram.
Intuition: $\kappa \to 1$ indicates strong, ordered gesture.

3.2 Curl and Divergence of Gesture Field

Construct a unit orientation field $\mathbf{u} = (\cos\theta, \sin\theta)$ from the dominant eigenvector of J .
Approximate **divergence** $\nabla \cdot \mathbf{u}$ and **curl** $(\nabla \times \mathbf{u})_z$ by finite differences.
Use: divergence ↔ fanning/centring; curl ↔ rotational torque/whorl.

3.3 Skeleton Graph & Branch Metrics

Binarize salient structures (e.g., via Frangi vesselness or Canny+thinning) → skeleton S .
Build a graph $G = (V, E)$ with node degree distribution, mean branch length, cycle count.
Use: anatomical “armature” of the image; articulation vs. tangle.

3.4 Persistence Across Thresholds (Barcode of Forms)

Define the sublevel sets of L as thresholds t sweep 0→255. Track connected components’ births and deaths → **persistence diagram** and **barcode**.
Use: emphasizes long-lived shapes (structural motifs) over noise.

3.5 Fractal Dimension and Lacunarity

Box-counting dimension **D** via multi-scale counts $N(\epsilon)$ and slope of $\log N$ vs $\log(1/\epsilon)$.
Lacunarity Λ : gappiness/heterogeneity measured by windowed mass variance at scale.
Use: D ~ space-filling, Λ ~ distribution of voids.

3.6 Kolmogorov Proxy ($\hat{K}$)

Compute **lossless compressibility** proxies: PNG and WEBP sizes, dictionary sizes. Normalize for resolution.
Use: low $\hat{K}$ ↔ regularity/repetition; high $\hat{K}$ ↔ novelty/noise.

---

4. Aesthetic Chemistry (Chromatic & Material Metaphors)

Palette Extraction: k-means or K-medoid in perceptual space (OKLab).
Gamut Area: convex hull area of palette.
Complementary Tension Score: weighted distances between high-energy complementary axes.
Local Diffusion: convolve color channels with anisotropic kernels aligned to gradient orientation; estimate **chromatic diffusion rate**.
Pigment Mixing Operators: simulate subtractive blending (Kubelka–Munk proxy) to predict stability of overpaints and glazes; compute **mixability index** (variance contraction under mixing).

Readouts: hue flow field, palette barycenter drift, chroma–lightness stability, patch-level metamers (chemistry of “near matches”).

---

5. Aesthetic Thermodynamics (Order, Energy, Attractors)

Entropy at Scale: Shannon entropy of luminance and hue channels under Gaussian pyramid.
Exhaustion Curve: area under entropy–scale curve; slope changes indicate regime shifts (minimalism vs. ornament).
Temperature Field: local variance-at-scale; detect “hot” kinetic zones vs. cool plateaus.
Attractor Sketches: iterate simple operators (blur→sharpen→threshold); examine fixed points and cycles.
Use: how a work “wants to settle.”

---

6. Fusion Metric: Aesthetic Complexity Index (ACI)

Let standardized z-scores be denoted by $z[\cdot]$. Define a tunable blend:

$$\begin{aligned}\mathrm{ACI} = w_1\, z[D] + w_2\, z(\overline{\Lambda}) + w_3\, z(\text{persistence span}) + w_4\, z(\text{skeleton branching}) + \\ w_5\, z(\overline{\kappa}) + w_6\, z(\hat{K}) + w_7\, z(\text{entropy area}) + w_8\, z(\text{palette tension}).\end{aligned}$$

Weights w_i are user-adjustable with presets: **Minimal**, **Ornate**, **Organic**, **Architectural**. The UI displays per-term contributions and sensitivity bars.

7. Complexity Atlas (Diagnostic Maps)

A small-multiples grid (e.g., 2×4) of: - Coherence map - Curl map - Divergence map - Skeleton overlay - Persistence heat (lifespan weights projected back to image) - Fractal driver heatmap (per-region contribution to D) - Lacunarity map at chosen window - Entropy-at-scale map

Each map is clickable for full-screen inspection and crosshair readouts.

---

8. Data Model & File Formats

- **Project JSON** with versioning, raw metrics, per-map PNGs (base64), and ACI.
- **CSV export** for tabular metrics (rows per image, columns per feature).
- **Bundle**: `.aedna` (zip of JSON + assets) for reproducibility.

---

9. Software Architecture (Spec)

Front-end: Next.js 15, React, Tailwind, shadcn/ui, Framer Motion, Recharts.
Workers: Web Workers / WASM kernels for filters, skeletonization, box-counting, and persistence tracking.
Core modules: - `calc/gradients.ts` – Sobel/Scharr, structure tensor, coherence. - `calc/fields.ts` – orientation field, curl/divergence. - `calc/topology.ts` – threshold sweep, union–find components, barcodes. - `calc/skeleton.ts` – thinning, graphization. - `chem/palette.ts` – OKLab space, k-medoids, gamut hull. - `chem/mix.ts` – subtractive mixing proxy, diffusion. - `thermo/entropy.ts` – pyramid entropy, exhaustion curve, temperature field. - `fusion/aci.ts` – normalization, presets, radar charts.

UI patterns: - **Global Header** with aeDNA logo (home), branch bar (Calculus • Chemistry • Thermodynamics), current tool highlight. - **Detail Header** per page with tool name, preset selector, export buttons. - **Inspector Drawer** shows numeric readouts for hovered pixel/tile. - **Compare Mode**: 2-up or 3-up synchronized views.

---

10. Validation & Insight Generation

Benchmarks: - **Human study:** artists rate movement, density, spaciousness → correlate with κ, divergence, Λ. - **Style cohorts:** compare metric signatures across schools (e.g., Baroque vs. Minimalist) to test discriminability. - **Ablation:** remove each metric and test ACI stability and rank order shifts.

Interpretability protocols: - Always show **per-metric maps** next to the scalar; clicking a number highlights its spatial drivers.

---

11. Roadmap

Phase 1–3 (done): Image ingestion, gradients, vector overlays, enlarge-on-click UI.
Phase 4: Persistence barcodes + skeleton graphs (exportable).
Phase 5: Complexity Atlas + ACI with presets.
Phase 6: Multi-Scale Topology Lab (box-counting heatmaps, lacunarity maps).
Phase 7: Chemistry & Thermodynamics branches; palette kinetics; entropy/exhaustion.
Phase 8: Dataset mode, comparisons, and whitepaper PDF export.

---

12. Ethics & Limits

Metrics are aids, not verdicts. We forbid using aeDNA for automated aesthetic ranking of people or culturally sensitive artifacts. The system must keep provenance, allow opt-out from dataset storage, and clearly explain features.

---

13. Glossary

- **Coherence (κ):** Orientation order from structure tensor eigenvalues.
- **Curl / Divergence:** Rotational vs. fanning tendency of the orientation field.
- **Persistence:** Lifespan of components across thresholds; topological robustness.
- **Lacunarity (Λ):** Heterogeneity of gaps.
- **K̂:** Compressibility proxy for Kolmogorov complexity.
- **Exhaustion Curve:** Entropy vs. scale integral capturing ornamental load.

Appendix A — Upload-Ready Starter Code (Cursor)

Below is a minimal Next.js 15 scaffold with the new **branches** layout, global header, a home splash with logo + typewriter "enter aesthetic genome," and a starter Calculus page wired to placeholder metrics (ready to connect to your existing kernels).

> **How to use:** Create files as shown in the tree. If you already have an app, merge `components/`, `app/(branches)/`, and `lib/` content. Tailwind + shadcn assumed.

```
# Files/folders to add under your Next.js project root
app/
  layout.tsx
  page.tsx
  (branches)/
    layout.tsx
    calculus/page.tsx
    chemistry/page.tsx
```

```
    thermodynamics/page.tsx
  whitepaper/page.tsx
components/
  Header.tsx
  BranchTabs.tsx
  MetricCard.tsx
  AtlasGrid.tsx
  TypePulse.tsx
lib/
  metrics/mock.ts
  styles.ts
public/
  logo-aedna.svg
```

```tsx
// app/layout.tsx
import "./globals.css";
import { ReactNode } from "react";

export default function RootLayout({ children }: { children: ReactNode }) {
  return (
    <html lang="en" suppressHydrationWarning>
      <body className="min-h-screen bg-[#0D0D0F] text-zinc-200 antialiased">
        {children}
      </body>
    </html>
  );
}
```

```tsx
// components/TypePulse.tsx
import { useEffect, useState } from "react";

export default function TypePulse({ text }: { text: string }) {
  const [shown, setShown] = useState(0);
  useEffect(() => {
    const id = setInterval(() => setShown((s) => Math.min(s + 1, text.length)),
45);
    return () => clearInterval(id);
  }, [text]);
  return <span className="tracking-widest">{text.slice(0, shown)}</span>;
}
```

```tsx
// components/BranchTabs.tsx
"use client";
import Link from "next/link";
```

```tsx
import { usePathname } from "next/navigation";

const tabs = [
  { href: "/calculus", label: "Calculus" },
  { href: "/chemistry", label: "Chemistry" },
  { href: "/thermodynamics", label: "Thermodynamics" },
];

export default function BranchTabs() {
  const pathname = usePathname();
  return (
    <div className="flex gap-4 text-sm">
      {tabs.map((t) => {
        const active = pathname.startsWith(t.href);
        return (
          <Link key={t.href} href={t.href}
            className={`px-3 py-1 rounded-full border ${active ? "border-
cyan-400 text-cyan-300" : "border-zinc-700 text-zinc-400 hover:text-zinc-200"}`}
          >
            {t.label}
          </Link>
        );
      })}
    </div>
  );
}
```

```tsx
// components/Header.tsx
import Link from "next/link";
import BranchTabs from "@/components/BranchTabs";

export default function Header({ title }: { title?: string }) {
  return (
    <header className="sticky top-0 z-50 backdrop-blur supports-[backdrop-
filter]:bg-black/30 border-b border-zinc-800">
      <div className="max-w-6xl mx-auto px-4 py-3 flex items-center justify-
between">
        <Link href="/" className="flex items-center gap-2">
          <img src="/logo-aedna.svg" alt="aeDNA" className="h-6" />
          <span className="sr-only">Home</span>
        </Link>
        <BranchTabs />
        <div className="text-xs text-zinc-500 hidden sm:block">{title ?? ""}</
div>
      </div>
    </header>
```

```
  );
}
```

```
// app/page.tsx (Home)
import TypePulse from "@/components/TypePulse";
import Link from "next/link";

export default function Home() {
  return (
    <main className="min-h-screen grid place-items-center">
      <div className="text-center">
        <div className="mb-6">
          <img src="/logo-aedna.svg" alt="aeDNA" className="mx-auto h-28
opacity-90" />
        </div>
        <p className="text-zinc-400 text-xs uppercase">Aesthetic Genome Lab</p>
        <h1 className="mt-2 text-4xl font-light tracking-tight">aeDNA</h1>
        <div className="mt-8 text-zinc-400">
          <TypePulse text="enter aesthetic genome" />
        </div>
        <div className="mt-10">
          <Link href="/calculus" className="px-6 py-2 rounded-full border
border-zinc-700 hover:border-cyan-400 hover:text-cyan-300">Explore</Link>
        </div>
      </div>
    </main>
  );
}
```

```
// app/(branches)/layout.tsx
import { ReactNode } from "react";
import Header from "@/components/Header";

export default function BranchLayout({ children }: { children: ReactNode }) {
  return (
    <>
      <Header />
      <main className="max-w-6xl mx-auto px-4 py-8">{children}</main>
    </>
  );
}
```

```
// components/MetricCard.tsx
export default function MetricCard({ title, value, subtitle }: { title: string;
```

```
value: string | number; subtitle?: string }) {
  return (
    <div className="rounded-2xl border border-zinc-800 p-4 hover:border-
cyan-400/40 transition">
      <div className="text-xs uppercase text-zinc-500">{title}</div>
      <div className="text-2xl mt-1">{value}</div>
      {subtitle && <div className="text-xs text-zinc-500 mt-2">{subtitle}</div>}
    </div>
  );
}
```

```
// components/AtlasGrid.tsx
export default function AtlasGrid({ images }: { images: { src: string; label:
string }[] }) {
  return (
    <div className="grid md:grid-cols-4 sm:grid-cols-2 gap-4">
      {images.map((m) => (
        <figure key={m.label} className="rounded-xl overflow-hidden border
border-zinc-800">
          <img src={m.src} alt={m.label} className="w-full h-40 object-cover
cursor-pointer" />
          <figcaption className="text-xs text-zinc-400 p-2">{m.label}</
figcaption>
        </figure>
      ))}
    </div>
  );
}
```

```
// lib/metrics/mock.ts
export const mockMetrics = {
  coherenceMean: 0.67,
  curlMean: 0.05,
  divergenceMean: -0.03,
  skeletonBranches: 142,
  persistenceSpan: 38.4,
  fractalD: 1.73,
  lacunarityMean: 1.21,
  Khat: 0.58,
  entropyArea: 0.64,
  paletteTension: 0.41,
  ACI: 0.69,
};
```

```tsx
// app/(branches)/calculus/page.tsx
import MetricCard from "@/components/MetricCard";
import AtlasGrid from "@/components/AtlasGrid";
import Header from "@/components/Header";
import { mockMetrics as m } from "@/lib/metrics/mock";

export const metadata = { title: "Calculus — aeDNA" };

export default function CalculusPage() {
  return (
    <>
      <Header title="Calculus • Field & Topology" />
      <section className="space-y-8">
        <div className="grid md:grid-cols-4 sm:grid-cols-2 gap-4">
          <MetricCard title="Coherence (κ)" value={m.coherenceMean.toFixed(2)}
subtitle="Flow order" />
          <MetricCard title="Curl (avg)" value={m.curlMean.toFixed(2)}
subtitle="Rotational torque" />
          <MetricCard title="Divergence (avg)"
value={m.divergenceMean.toFixed(2)} subtitle="Fanning / centering" />
          <MetricCard title="Branches" value={m.skeletonBranches}
subtitle="Skeleton graph" />
          <MetricCard title="Persistence span"
value={m.persistenceSpan.toFixed(1)} subtitle="Barcode lifespan" />
          <MetricCard title="Fractal D" value={m.fractalD.toFixed(2)}
subtitle="Space-filling" />
          <MetricCard title="Lacunarity (Λ)"
value={m.lacunarityMean.toFixed(2)} subtitle="Gap heterogeneity" />
          <MetricCard title="K̂" value={m.Khat.toFixed(2)}
subtitle="Compressibility" />
        </div>
        <div>
          <h3 className="text-sm uppercase text-zinc-500 mb-3">Complexity
Atlas</h3>
          <AtlasGrid
            images={[
              { src: "/placeholder/coherence.png", label: "Coherence" },
              { src: "/placeholder/curl.png", label: "Curl" },
              { src: "/placeholder/divergence.png", label: "Divergence" },
              { src: "/placeholder/skeleton.png", label: "Skeleton" },
              { src: "/placeholder/persistence.png", label: "Persistence" },
              { src: "/placeholder/fractal.png", label: "Fractal Driver" },
              { src: "/placeholder/lacunarity.png", label: "Lacunarity" },
              { src: "/placeholder/entropy.png", label: "Entropy@Scale" },
            ]}
          />
        </div>
```

```
        <div className="grid md:grid-cols-3 gap-4">
          <MetricCard title="Entropy Area" value={m.entropyArea.toFixed(2)}
subtitle="Order vs. ornament" />
          <MetricCard title="Palette Tension"
value={m.paletteTension.toFixed(2)} subtitle="Complementary energy" />
          <MetricCard title="ACI" value={m.ACI.toFixed(2)} subtitle="Aesthetic
Complexity Index" />
        </div>
      </section>
    </>
  );
}
```

```
// app/(branches)/chemistry/page.tsx
import Header from "@/components/Header";

export default function ChemistryPage() {
  return (
    <>
      <Header title="Chemistry • Palette & Material" />
      <p className="text-zinc-400">Palette extraction, gamut area,
complementary tension, pigment mixing operators — wiring to follow.</p>
    </>
  );
}
```

```
// app/(branches)/thermodynamics/page.tsx
import Header from "@/components/Header";

export default function ThermoPage() {
  return (
    <>
      <Header title="Thermodynamics • Order & Energy" />
      <p className="text-zinc-400">Entropy@scale, exhaustion curves,
temperature fields, attractor sketches — wiring to follow.</p>
    </>
  );
}
```

```
// app/whitepaper/page.tsx
import Header from "@/components/Header";

const md = `# Aesthetic DNA White Paper\nThis route mirrors the canonical white
paper. For PDF export, pipe this markdown through your printer or a serverless
```

```
PDF function.`;

export default function WhitepaperPage() {
  return (
    <>
      <Header title="White Paper" />
      <article className="prose prose-invert max-w-none">{md}</article>
    </>
  );
}
```

```
// lib/styles.ts
export const card = "rounded-2xl border border-zinc-800 p-4";
```

Notes: - Replace `/placeholder/*.png` with your generated atlas tiles. - Swap `mockMetrics` for real kernels. Each metric card should be tied to a selector that, when clicked, highlights its map. - Global header and branch tabs match your requested subpage header and home button.

Appendix B — Metric Definitions (Concise)

- **Coherence κ:** `(λ1−λ2)/(λ1+λ2+ε)` from structure tensor; mean and histogram.
- **Curl:** finite difference on unit orientation field; signed average and map.
- **Divergence:** ditto; signed average and map.
- **Skeleton Graph:** nodes (junctions/endpoints), edges (branches), cycles; degree histogram; mean branch length.
- **Persistence Span:** mean/quantile lifespan of components in threshold sweep.
- **Fractal D:** slope of log–log box count; regional contributions heatmap.
- **Lacunarity Λ:** mass variance in sliding windows at chosen scales.
- **K̂:** normalized PNG/WEBP size; optional LZ dictionary readouts.
- **Entropy@Scale:** Shannon entropy across pyramid; integral = exhaustion area.
- **Palette Tension:** complementary-axis energy in OKLab; local field option.
- **ACI:** weighted blend with presets, slider-exposed weights.

End of Document