Appendix for Lenia: Biology of Artificial Life

Bert Wang-Chak Chan
Hong Kong

This is the appendix for the paper that describes Lenia, an artificial life system of a two-dimensional cellular automaton with continuous space-time state and generalized local rule. This appendix includes (A) a guide to computer implementation; (B) more results, including the tree of artificial life, architecture and symmetry; (C) a case study of the life form genus *Paraptera*; (D) more discussion on the nature of Lenia; and (E) open questions and future work.

Keywords: artificial life; geometric cellular automata; complex system; interactive evolutionary computation

A. Computer Implementation

Discrete Lenia can be implemented with the following pseudocode, assuming an array programming language is used (e.g., Python with NumPy, MATLAB, Wolfram Language).

Interactive programs have been written in JavaScript/HTML5, Python and MATLAB to provide a user interface for species discovery and real-time analysis (Figure A.1(a–b)). A noninteractive program has been written in C#.NET for automatic traverse through the parameter space using a flood fill algorithm (breadth-first or depth-first search), providing species distribution, statistical data and occasionally new species.

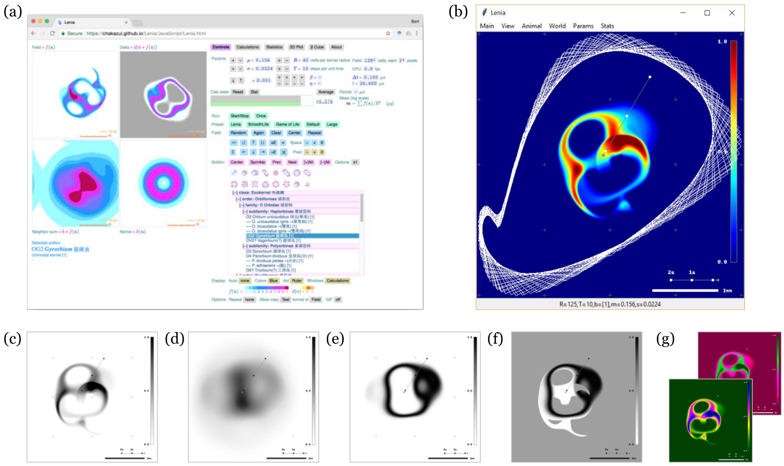


Figure A.1. Computer implementations of Lenia with interactive user interfaces. (a) Web version run in Chrome browser and (b) Python version with GPU support. (c–f) Different views during simulation, including (c) the configuration $\mathbf{A}^t$, (d) the potential $\mathbf{U}^t$, (e) the growth $\mathbf{G}^t$ and (f) the actual change $\Delta\mathbf{A} / \Delta t = \left(\mathbf{A}^{t+\Delta t} - \mathbf{A}^t\right) / \Delta t$. (g) Other color schemes.

State precision Δp can be implicitly implemented as the precision of floating-point numbers. For values in the unit interval $[0, 1]$, the precision ranges from 2^{-126} to 2^{-23} (about 1.2×10^{-38} to 1.2×10^{-7}) using 32-bit single precision, or from 2^{-1022} to 2^{-52} (about 2.2×10^{-308} to 2.2×10^{-16}) using 64-bit double precision [1]. That means $P > 10^{15}$ using double precision.

Discrete convolution can be calculated as the sum of elementwise products:

$$\mathbf{K} * \mathbf{A}^t(\mathbf{x}) = \sum_{\mathbf{n}\in\mathcal{N}} \mathbf{K}(\mathbf{n})\mathbf{A}^t(\mathbf{x}+\mathbf{n}) \tag{A.1}$$

or alternatively, using the discrete Fourier transform (DFT) according to the convolution theorem:

$$\mathbf{K} * \mathbf{A}^t = F^{-1}\left\{F\{\mathbf{K}\} \cdot F\left\{\mathbf{A}^t\right\}\right\}. \tag{A.2}$$

Efficient calculation can be achieved using the fast Fourier transform (FFT) [2], pre-calculation of the kernel's FFT $\mathcal{F}\{\mathbf{K}\}$ and parallel computing like GPU acceleration. The DFT/FFT approach automatically produces a periodic boundary condition.

A.1 Pseudocode

The symbol @ indicates a two-dimensional matrix of floating-point numbers.

```
function pre_calculate_kernel(beta, dx)
 @radius = get_polar_radius_matrix(SIZE_X, SIZE_Y) * dx
 @Br = size(beta) * @radius
 @kernel_shell = beta[floor(@Br)] * kernel_core(@Br % 1)
 @kernel = @kernel_shell / sum(@kernel_shell)
 @kernel_FFT = FFT_2D(@kernel)
 return @kernel, @kernel_FFT
end

function run_automaton(@world, @kernel, @kernel_FFT, mu, sigma, dt)
 if size(@world) is small
  @potential = elementwise_convolution(@kernel, @world)
 else
  @world_FFT = FFT_2D(@world)
  @potential_FFT = elementwise_multiply(@kernel_FFT, @world_FFT)
  @potential = FFT_shift(real_part(inverse_FFT_2D(@potential_FFT)))
 end
 @growth = growth_mapping(@potential, mu, sigma)
 @new_world = clip(@world + dt * @growth, 0, 1)
 return @new_world, @growth, @potential
end

function simulation()
 R, T, mu, sigma, beta = get_parameters()
 dx = 1/R;  dt = 1/T;  time = 0
 @kernel, @kernel_FFT = pre_calculate_kernel(beta, dx)
 @world = get_initial_configuration(SIZE_X, SIZE_Y)
 repeat
  @world, @growth, @potential = run_automaton(@world,
    @kernel, @kernel_FFT, mu, sigma, dt)
  time = time + dt
  display(@world, @potential, @growth)
 end
end
```

A.2 User Interface

For implementations requiring an interactive user interface, one or more of the following components are recommended:

- Controls for starting and stopping CA simulation
- Panels for displaying different stages of CA calculation
- Controls for changing parameters and spacetime-state resolutions

- Controls for randomizing, transforming and editing the configuration
- Controls for saving, loading and copy-and-pasting configurations
- Clickable list for loading predefined patterns
- Utilities for capturing the display output (e.g., image, GIF, movie)
- Controls for customizing the layout (e.g., grid size, color map)
- Controls for autocentering, autorotating and temporal sampling
- Panels or overlays for displaying real-time statistical analysis

A.3 Pattern Storage

A pattern can be stored for publication and sharing using a data exchange format (e.g., JSON, XML) that includes the *run-length encoding* (RLE) of the two-dimensional array $\mathbf{A}^t$ and its associated settings $(R, T, P, \mu, \sigma, \beta, K_C, G)$, or alternatively, using a plaintext format (e.g., CSV) for further analysis or manipulation in numeric software.

A long list of interesting patterns can be saved as JSON/XML for program retrieval. To save storage space, patterns can be stored with space resolution R as small as possible (usually $10 \leq R \leq 20$) thanks to Lenia's scale invariance (see Section 3.1).

A.4 Environment

Most of the computer simulations, experiments, statistical analysis, image and video capturing for this paper were done using the following environments and settings:

- Hardware: Apple MacBook Pro (OS X Yosemite), Lenovo ThinkPad X280 (Microsoft Windows 10 Pro)
- Software: Python 3.7.0, MathWorks MATLAB Home R2017b, Google Chrome browser, Microsoft Excel 2016
- State precision: double precision
- Kernel core and growth mapping: exponential

B. More Results

B.1 Tree of Artificial Life

The notion of "life," here interpreted as self-organizing autonomous entities in a broader sense, may include biological life, artificial life and other possibilities like extraterrestrial life. Based on life forms from Lenia and other systems, we propose the *tree of artificial life*:

Artificialia
Domain Synthetica, “wet” biochemical synthetic life
Domain Mechanica, “hard” mechanical or robotic life, e.g., [3]
Domain Simulata, “soft” computer simulated life
Kingdom Sims, evolved virtual creatures, e.g., [4–6]
Kingdom Greges, particle swarm solitons, e.g., [7–10]
Kingdom Turing, reaction-diffusion solitons, e.g., [11–13]
Kingdom Automata, cellular automata solitons
Phylum Discreta, nonscalable, e.g., [14–16]
Phylum Lenia, scalable, e.g., [17, 18]

The current taxonomy of Lenia (Figure 7):
Phylum Lenia
Class Exokernel having strong outer kernel rings
Order Orbiformes
Family Orbidae (O), “disk bugs,” disks with central stalk
Order Scutiformes
Family Scutidae (S), “shield bugs,” disks with thick front
Family Pterifera (P), “winged bugs,” one/two wings with sacs
Family Helicidae (H), “helix bugs,” rotating versions of P
Family Circidae (C), “circle bugs,” one or more concentric rings
Class Mesokernel having kernel rings of similar heights
Order Echiniformes
Family Echinidae (E), “spiny bugs,” thorny or wavy species
Family Geminidae (G), “twin bugs,” two or more compartments
Family Ctenidae (Ct), “comb bugs,” P with narrow strips
Family Uridae (U), “tailed bugs,” with tails of various lengths
Class Endokernel having strong inner kernel rings
Order Kroniformes
Family Kronidae (K), “crown bugs,” complex versions of S, P
Family Quadridae (Q), “square bugs,” 4×4 grids of masses
Family Volvidae (V), “twisting bugs,” possibly complex H
Order Radiiformes
Family Dentidae (D), “gear bugs,” rotating with gear-like units
Family Radiidae (R), “radial bugs,” regular or star polygon–shaped
Family Bullidae (B), “bubble bugs,” bilateral with bubbles inside
Family Lapillidae (L), “gem bugs,” radially distributed small rings
Family Folidae (F), “petal bugs,” stationary with petal-like units
Order Amoebiformes
Family Amoebidae (A), “amoeba bugs,” volatile shape and behavior

Much like real-world biology, the taxonomy of Lenia is tentative and is subject to revisions or redefinitions when more data is available.

B.2 Naming

Following Jansen for naming artificial life using biological nomenclature (*Animaris* spp.) [3], each Lenia species was given a binomial name that describes its geometric shape (genus name) and behavior (species name) to facilitate analysis and communication. Alphanumeric code was given in the form "*BGUs*" with initials of rank (B), genus or family name (G), species name (s) and number of units (U).

The suffix "*-ium*" in genus names is reminiscent of a bacterium or chemical elements, while suffixes "-inae" (subfamily), "-idae" (family) and "-iformes" (order) were borrowed from actual animal taxa. The numeric prefix in genus names indicates the number of units, similar to organic compounds and elements (IUPAC names). Prefixes used are: *Di-, Tri-, Tetra-, Penta-, Hexa-, Hepta-, Octa-, Nona-, Deca-, Undeca-, Dodeca-, Trideca-* and so on.

B.3 Architecture

Lenia life forms possess morphological structures of various kinds, but they can be summarized into the following types of architectures:

- *Segmented architecture* is the serial combination of a few basic components, prevalent in class Exokernel (O, S, P, H), also Ct, U, K.
- *Radial architecture* is the radial arrangement of repeating units, common in Radiiformes in class Endokernel (D, R, B, L, F), also C, E, V.
- *Swarm architecture* is the volatile cluster of granular masses, not confined to a particular geometry or locomotion, as in G, Q, A.

B.4 Components and Metamerism

Segmented architecture is composed of the following inventory of components (class Exokernel only) (Figure 10(a–c, f)).

- The *orb* (disk) is a circular disk halved by a central stalk, found in O.
- The *scutum* (shield) is a disk with a thick front shield, found in S.
- The *wing* has two versions: the *orboid* (disk-like) wing is a distorted orb with a budding mechanism that creates and destroys sacs repeatedly, found in concave S, P, H; the *scutoid* (shield-like) wing is a distorted scutum, found in convex S, P, H.
- The *vacuole* (sac) is a disk between the wings of long-chain S, P, H.

Many of these components are possibly interrelated, for example, the orboid wing and the orb, the scutoid wing and the scutum, as suggested by the similarity or smooth transitions between species.

Multiple components can be combined serially into long chains through fusion or adhesion (e.g., Figure 7(O:2) or (O:1)), in a fashion comparable to *metamerism* in biology (or *multicellularity* if we consider the components as "cells") (Figure 10(f–g)).

Long-chain species exhibit different degrees of *convexity*, from convex to concave: S > convex P (*arcus* subgenus) > linear O > concave P (*cavus* subgenus); sinusoidal P (*sinus* subgenus) have hybrid convexity (Figure 10(i), 7 column 1).

Higher-rank segmented Ct, U, K also exhibit metamerism and convexity with more complicated components.

B.5 Symmetry and Asymmetry

Structural symmetry is a prominent characteristic of Lenia life, including the following types:

- *Bilateral symmetry* (dihedral group D_1) mostly in segmented and swarm architectures (O, S, P, Ct, U, K; G, Q).
- *Radial symmetry* (dihedral group D_n) is geometrically rotational plus reflectional symmetry, caused by bilateral repeating units in radial architecture (R, L, F, E).
- *Rotational symmetry* (cyclic group C_n) is geometrically rotational without reflectional symmetry, caused by asymmetric repeating units in radial architecture (D, R, L) (Figure 10(h)).
- *Spherical symmetry* (orthogonal group O(2)) is a special case of radial symmetry (C).
- Secondary symmetries:
  - *Spiral symmetry* is secondary rotational symmetry derived from twisted bilaterals (H, V).
  - *Biradial symmetry* is secondary bilateral symmetry derived from radials (B, R, E).
  - *Deformed bilateral symmetry* is bilateral with heavy asymmetry (e.g., gyrating species in O, S, G, Q).
- No symmetry in amorphous species (A).

Asymmetry also plays a significant role in shaping the life forms and guiding their movements, causing various degrees of angular motions (detailed in Section 3.6). Asymmetry is usually intrinsic in a species, as demonstrated by experiments where a slightly asymmetric form (e.g., *Paraptera pedes*, *Echinium limus*) was mirrored into perfect symmetry and remained metastable, but after the slightest perturbation (e.g., rotate 1°), it slowly restores to its natural asymmetric form.

B.6 Ornamentation

Many detailed local patterns arise in higher-rank species, owing to their complex kernels (Figure 10(d–e, j–l)):

- *Decoration* is the addition of tiny ornaments (e.g., dots, circles, crosses), prevalent in class Endokernel.
- *Serration* is a ripple-like sinusoidal boundary or pattern, common in classes Exokernel and Mesokernel.

- *Caudation* is a tail-like structure behind a long-chain life form (e.g., P, K, U), akin to “tag-along” in GoL.
- *Liquefaction* is the degradation of an otherwise regular structure into a chaotic “liquified” tail.

C. Case Study

In previous sections, we outlined the general characterizations of Lenia from various perspectives. Here we combine these aspects in a focused study of one representative genus—*Paraptera* (P4)—as a demonstration of concrete qualitative and quantitative analysis.

C.1 The Unit-4 Group

Paraptera (P4) is closely related to two other genera, *Parorbium* (O4) and *Tetrascutium* (S4); they comprise the rank-1, unit-4 group.

In the μ–σ map (Figure C.1), their niches comprise the *Parorbium-Paraptera-Tetrascutium* (O4-P4-S4) complex. The narrow bridge between O4 and P4 indicates possible continuous transformation, and the agreement between the small tip of P4 and S4 suggests a remote relationship. Species were isolated using allometric methods (Figure C.2, Table 4), verified in simulation and assigned new names (Table C.1).

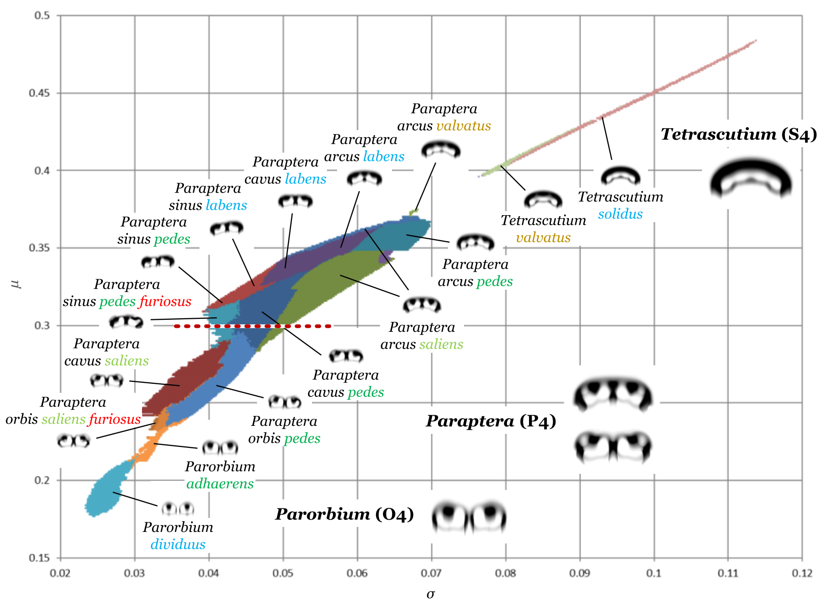


Figure C.1. μ–σ map of the unit-4 group, showing the prominent *Parorbium-Paraptera-Tetrascutium* complex. Total 16 011 loci. The red dotted line marks the cross-sectional study (Figure C.3).

Species			Morphology	Behavior
O4	Genus *Parorbium* (Family Orbidae)			
	O4d	*Po. dividuus*	Two parallel orbs, separated	T = translocating
	O4a	*Po. adhaerens*	Two parallel orbs, adhered	T
P4	Genus *Paraptera* (Family Pterifera)			
	P4o∗	*P. orbis* ∗	Concave, twin orboid wings	T
	P4c∗	*P. cavus* ∗	Concave, twin orboid wings	T
	P4a∗	*P. arcus* ∗	Convex, twin scutoid wings	T
	P4s∗	*P. sinus* ∗	Sinusoidal, orboid + scutoid wings	T_{D*} = deflected
	P4∗l	*P.* ∗ *labens*	Bilateral	T_F = sliding
	P4∗s	*P.* ∗ *saliens*	Bilateral	T_O = jumping
	P4∗p	*P.* ∗ *pedes*	Bilateral with slight asymmetry	T_A = walking
	P4∗v	*P.* ∗ *valvatus*	Scutidae-like, twin wings, valving	T_O = valving
	P4∗∗f	*P.* ∗ ∗ *furiosus*	Occasional stretched wing	T_{C*} = chaotic
S4	Genus *Tetrascutium* (Family Scutidae)			
	S4s	*T. solidus*	Four fused scuta, solid	T_F = sliding
	S4v	*T. valvatus*	Four fused scuta, valving	T_O = valving

Table C.1. Non-exhaustive list of species identified in the unit-4 group. (* = combinations are possible, e.g., P4spf with behavior T_{CDA}).

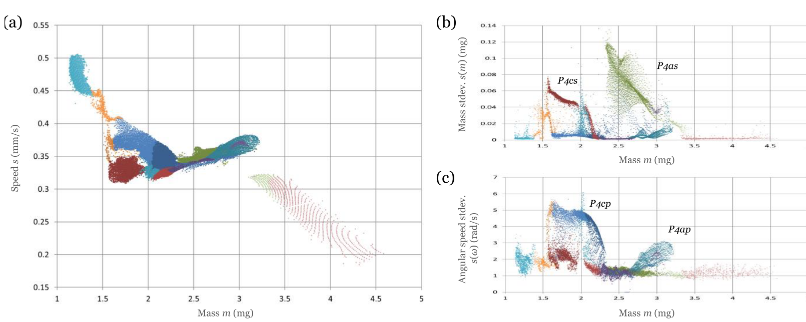


Figure C.2. Allometric charts of various measures for the unit-4 family. Total 16 011 loci, 300 time steps ($t = 30$s) per locus. (a) Linear speed s_m versus mass m, similar to the μ–σ map flipped. (b) Mass variability $s(m)$ versus mass m, isolates jumping (T_O) species P4cs and P4as. (c) Angular speed variability $s(\omega_m)$ versus mass m, isolates walking (T_A) species P4cp and P4ap.

C.2 Cross-Sectional Study

In P4, a cross section at $\mu = 0.3$ was further investigated, where five species exist in $\sigma \in [0.0393, 0.0515]$ (Figure C.1 red dotted line, Table C.2). Their behavioral traits were assessed via cross-sectional charts and snapshot phase space trajectories (Figure C.3, see also Figure 11(e–i)).

At higher σ values, *Paraptera arcus saliens* (P4as) has high m variability and near zero m_Δ, corresponding to its jumping behavior and perfect bilateral symmetry (locus a). *P. cavus pedes* (P4cp) has high m_Δ variability, matching its walking behavior and alternating asymmetry (locus d).

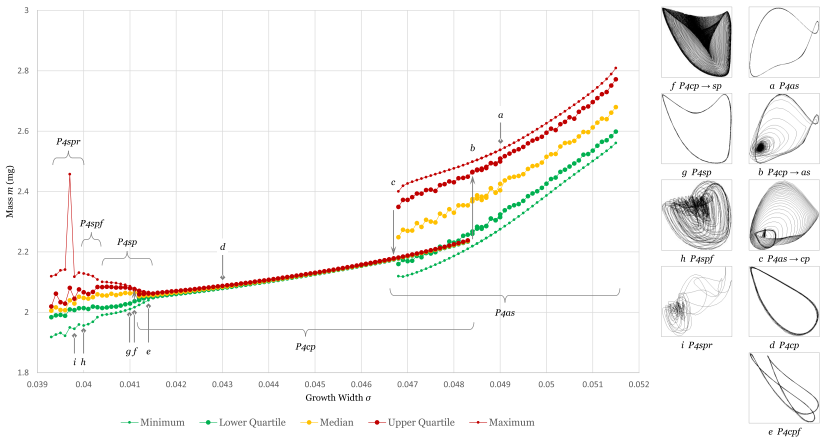
Just outside the coexistence of P4as and P4cp over $\sigma \in [0.0468, 0.0483]$, they slowly transform into each other, as shown by the spiral phase space trajectories (loci b, c). Similarly for P4cp and P4sp (locus f).

Irregularity and chaos arise at lower σ. For *P. sinus pedes* (P4sp), nonzero m_Δ indicates deflected movement and asymmetry (locus g). For *P. sinus pedes furiosus* (P4spf), chaotic phase space trajectory indicates chaotic movement and deformation (locus h).

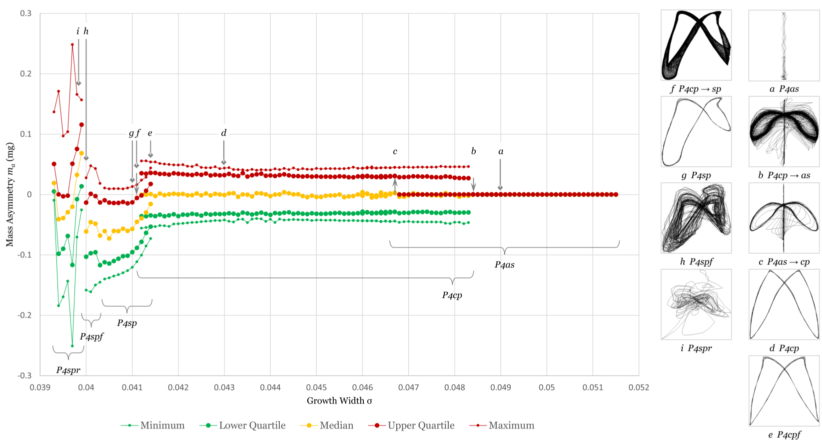
At the edge of chaoticity, *P. sinus pedes rupturus* (P4spr) has even higher and more rugged variability and often encounters episodes of acute deformation but eventually recovers (locus i). Outside the σ lower bound, the pattern fails to recover and finally disintegrates.

Species		σ Range	Morphology and Behavior
P4as	*P. arcus saliens*	[0.0468, 0.0515]	convex, jumping (T_O)
P4cp	*P. cavus pedes*	[0.0412, 0.0483]	concave, walking (T_A)
P4sp	*P. sinus pedes*	[0.0404, 0.0414]	sinusoidal, deflected walking (T_{DA})
P4spf	*P. s. p. furiosus*	[0.0400, 0.0403]	like previous plus chaotic (T_{CDSA})
P4spr	*P. s. p. rupturus*	[0.0393, 0.0399]	like previous plus fragile

Table C.2. List of *Paraptera* species in the cross section $\mu = 0.3$.



(a)



(b)

Figure C.3. Cross-sectional charts at $\mu = 0.3$, $\sigma \in [0.0393, 0.0515]$ in genus *Paraptera*. 200 time steps ($t = 20$s) per locus (see Table C.2 for species codes). (a) Mass m versus parameter σ, insets: growth g versus mass m phase space trajectories at loci a–i. (b) Mass asymmetry m_Δ versus parameter σ, insets: linear speed s_m versus angular speed ω_m phase space trajectories at loci a–i.

D. More on the Nature of Lenia

D.1 Quasi Periodicity

Unlike GoL, where a recurrent pattern returns to the exact same pattern after an exact period of time, a recurrent pattern in Lenia returns to similar patterns after slightly irregular periods or *quasi periods*, probably normally distributed. Lenia has various types of periodicity:

1. Aperiodic: in transient non-recurrent patterns
2. Quasi periodic: in quasi stable, stable or metastable patterns
3. Chaotic: with widespread quasi period distribution
4. Markovian: each template has its own type of periodicity

Principally, in discrete Lenia, there are a finite, albeit astronomically large, number of possible configurations $\left|S^{\mathcal{L}}\right|$. Given enough time, a recurrent pattern would eventually return to the exact same configuration (strictly periodic), an argument not unlike Nietzsche's "eternal recurrence," although there would be numerous approximate recurrences between two exact recurrences. In continuous Lenia, exact recurrence may even be impossible.

D.2 Plasticity

Given the fuzziness and irregularity, Lenia patterns are surprisingly resilient and exhibit *phenotypic plasticity*. By elastically adjusting morphology and behavior, they are able to absorb deformations and transformations, adapt to environmental changes (parameters and rule settings), react to head-to-head collisions and continue to survive.

We propose a speculative mechanism for the plasticity (also self-organization and self-regulation in general) as the *kernel resonance hypothesis* (Figure D.1). A network of *potential peaks* can be observed in the potential distribution. The peaks are formed by the overlapping or "resonance" of kernel rings cast by various mass lumps; in turn, the locations of the mass lumps are determined by the network of peaks. In this way, the mass lumps influence each other reciprocally and self-organize into structures, providing the basis of morphogenesis.

Kernel resonance is dynamic over time and may even be self-regulating, providing the basis of homeostasis. Plasticity may stem from the static buffering and dynamical flexibility provided by such mass-potential-mass feedback loop.

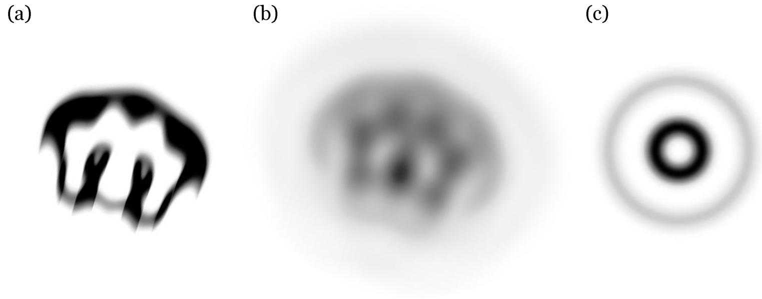


Figure D.1. Different views of calculation intermediates. (a) Configuration $\mathbf{A}^t$, (b) potential distribution $\mathbf{U}^t$, and (c) kernel $\mathbf{K}$. Notice one larger and six smaller potential peaks (b: dark spots) possibly formed by kernel resonance, and the corresponding inner spaces (a: white areas).

D.3 Computability

GoL, ECA rule 110, and LtL have been demonstrated to be capable of universal computation [16, 19, 20]. The proof of Turing universality of a CA requires searching for "glider gun" patterns that periodically emit "gliders" (i.e., small moving solitons), designing precise circuits orchestrated by glider collisions and assembling them into logic gates, memory registers and eventually Turing machines [21]. However, this may be difficult in Lenia due to the imprecise nature of pattern movements and collisions, and the lack of pattern-emitting configurations.

That said, particle collisions in Lenia, especially among *Orbium* instances, are worth further experimentation and analysis. These have been done for classical CAs (GoL and ECA rule 100) qualitatively [22] and quantitatively using, for example, algorithmic information dynamics [23].

E. Future Work

E.1 Open Questions

Here are a few open questions we hope to answer:

1. What are the enabling factors and mechanisms of how self-organization, self-regulation, self-direction, adaptability and so on emerge in Lenia?
2. How do interesting phenomena like symmetry, alternation, metamerism, metamorphosis, particle collision and so on arise in Lenia?
3. How is Lenia related to biological life and other forms of artificial life?

4. Can Lenia life be classified objectively and systematically?
5. Does continuous Lenia exist as the continuum limit of discrete Lenia? If so, do corresponding "ideal" life forms exist there?
6. Is Lenia Turing-complete and capable of universal computation?
7. Is Lenia capable of open-ended evolution that generates unlimited novelty and complexity?
8. Do self-replicating and pattern-emitting life forms exist in Lenia?
9. Do life forms exist in other variants of Lenia (e.g., three dimensional)?

To answer these questions, the following approaches to future work are suggested.

E.2 More Species Data

For the sheer joy of discovering new species and for further understanding Lenia and artificial life, we need better capabilities in species discovery and identification.

Automatic and accurate species identification could be achieved via computer vision and pattern recognition using machine learning or deep learning techniques, for example, training convolutional neural networks (CNNs) with patterns, or recurrent neural networks (RNNs) with time series of measures.

Interactive evolutionary computation (IEC) currently in use for new species discovery could be advanced to allow crowdsourcing. Web or mobile applications with an intuitive interface would allow online users to simulate, mutate, select and share interesting patterns (cf., Picbreeder [24], Ganbreeder [25]). Web performance and functionality could be improved using WebAssembly, OpenGL, TensorFlow.js and other programs.

Alternatively, evolutionary computation (EC) and similar methodologies could be used for automatic, efficient exploration of the search space, as has been successfully used for evolving new body parts or body plans [6, 3, 26]. Patterns could be represented in genetic (indirect) encoding using a compositional pattern-producing network (CPPN) [27] or Bezier splines [28], which are then evolved using genetic algorithms like NeuroEvolution of Augmenting Topologies (NEAT) [29]. Novelty-driven and curiosity-driven algorithms are promising approaches [30–32].

E.3 Better Data Analysis

Grid traversal of the parameter space (depth-first or breadth-first search) is still useful in collecting statistical data, but it needs more reliable algorithms, especially for high-rank metamorphosis-prone species.

All data collected from automation or crowdsourcing would be stored in a central database for further analysis. Using well-established techniques in related scientific disciplines, the data could be used for dynamical systems analysis (e.g., quasi period distribution, Lyapunov exponents, transition probabilities matrix), shape analysis (computational anatomy, statistical shape analysis, algorithmic complexity [33]), time-series analysis (cf., in astronomy [34]) and automatic classification (unsupervised or semi-supervised learning).

E.4 Variants and Generalizations

We could also explore variants and further generalizations of Lenia, for example, higher-dimensional spaces (e.g., three dimensional) [13, 35, 36]; different kinds of grids (e.g., hexagonal, Penrose tiling, irregular mesh) [37–39]; different structures of kernel (e.g., non-concentric rings); and other updating rules (e.g., asynchronous, heterogeneous, stochastic) [40–42].

E.5 Artificial Life and Artificial Intelligence

It has been demonstrated that Lenia shows a few signs of a living system:

- Self-organization: patterns develop well-defined structures.
- Self-regulation: patterns maintain dynamical equilibria via oscillation.
- Self-direction: patterns move consistently through space.
- Adaptability: patterns adapt to changes via plasticity.
- Evolvability: patterns evolve via manual operations and potentially genetic algorithms.

We should investigate whether these are merely superficial resemblances with biological life or are indications of deeper connections. In the latter case, Lenia could contribute to the endeavors of artificial life in attempting to "understand the essential general properties of living systems by synthesizing life-like behavior in software" [43], or could even add to the debate about the definitions of life as discussed in astrobiology and virology [44, 45]. In the former case, Lenia can still be regarded as a "mental exercise" on how to study a complex system using various methodologies.

Lenia could also serve as a "machine exercise" to provide a substrate or test bed for parallel computing, artificial life and artificial intelligence. The heavy demand in matrix calculation and pattern recognition could act as a benchmark for machine learning and hardware acceleration; the huge search space of patterns, possibly in higher dimensions, could act as a playground for evolutionary algorithms in the quest of algorithmizing and ultimately understanding open-ended evolution [46].

References

[1] W. Kahan. “Lecture Notes on the Status of IEEE Standard 754 for Binary Floating-Point Arithmetic.” (Aug 30, 2019) studfiles.net/preview/429637.

[2] J. W. Cooley and J. W. Tukey, “An Algorithm for the Machine Calculation of Complex Fourier Series,” *Mathematics of Computation*, **19**(90), 1965 pp. 297–301. doi:10.1090/S0025-5718-1965-0178586-1.

[3] T. Jansen, “Strandbeests,” *Architectural Design*, **78**(4), 2008 pp. 22–27. doi:10.1002/ad.701.

[4] K. Sims, “Evolving 3D Morphology and Behavior by Competition,” *Artificial Life*, **1**(4), 1994 pp. 353–372. doi:10.1162/artl.1994.1.4.353.

[5] N. Cheney, R. MacCurdy, J. Clune and H. Lipson, “Unshackling Evolution: Evolving Soft Robots with Multiple Materials and a Powerful Generative Encoding,” in *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation (GECCO ’13)*, Amsterdam, Netherlands, 2013, New York: ACM, 2013 pp. 167–174. doi:10.1145/2463372.2463404.

[6] S. Kriegman, N. Cheney and J. Bongard, “How Morphological Development Can Guide Evolution,” *Scientific Reports*, **8**(1), 2018 13934. doi:10.1038/s41598-018-31868-7.

[7] T. Schmickl, M. Stefanec and K. Crailsheim, “How a Life-Like System Emerges from a Simple Particle Motion Law,” *Scientific Reports*, **6**, 2016 37969. doi:/10.1038/srep37969.

[8] H. Sayama, “Swarm Chemistry Homepage: Sample Recipes.” (Aug 29, 2019) bingweb.binghamton.edu/~sayama/SwarmChemistry/#recipes.

[9] H. Sayama, “Seeking Open-Ended Evolution in Swarm Chemistry,” in *2011 IEEE Symposium on Artificial Life (ALIFE)*, Paris, France, 2011, Piscataway, NJ: IEEE, 2011 pp. 186–193. doi:10.1109/ALIFE.2011.5954667.

[10] H. Sayama, “Seeking Open-Ended Evolution in Swarm Chemistry II: Analyzing Long-Term Dynamics via Automated Object Harvesting,” in *ALIFE 2018: The 2018 Conference on Artificial Life* (T. Ikegami, N. Virgo, O. Witkowski, M. Oka, R. Suzuki and H. Izuka, eds.), Tokyo, Japan, 2018, The MIT Press, 2018 pp. 59–66. doi:10.1162/isal_a_ 00018.

[11] R. P. Munafo, “Stable Localized Moving Patterns in the 2-D Gray–Scott Model.” arxiv.org/abs/1501.01990.

[12] R. P. Munafo, “Catalog of Patterns at $F = 0.0620$, $k = 0.0609$.” (Aug 29, 2019) mrob.com/pub/comp/xmorphia/catalog.html.

[13] T. J. Hutton. *A 3D Glider in the U-Skate World (Reaction-Diffusion)* [Video]. (Aug 30, 2019) www.youtube.com/watch?v=WYZVffOaRgA.

[14] A. Adamatzky, ed., *Game of Life Cellular Automata*, New York: Springer, 2010.

[15] ConwayLife.com. "LifeWiki, the Wiki for Conway's Game of Life." (Aug 29, 2019) www.conwaylife.com/wiki/Main_Page.

[16] M. Cook, "Universality in Elementary Cellular Automata," *Complex Systems*, **15**(1), 2004 pp. 1–40. complex-systems.com/pdf/15-1-1.pdf.

[17] K. M. Evans, "Larger than Life: Digital Creatures in a Family of Two-Dimensional Cellular Automata," in *Discrete Mathematics and Theoretical Computer Science Proceedings, Vol. AA*, 2001 pp. 177–192. www.emis.ams.org/journals/DMTCS/pdfpapers/dmAA0113.pdf.

[18] S. Rafler, "Generalization of Conway's "Game of Life" to a Continuous Domain—SmoothLife." arxiv.org/abs/1111.1567.

[19] P. Rendell, "Turing Universality of the Game of Life," in *Collision-Based Computing* (A. Adamatzky, ed.), London: Springer, 2002 pp. 513–539. doi:10.1007/978-1-4471-0129-1_18.

[20] K. M. Evans, "Is Bosco's Rule Universal?," in *Machines, Computations, and Universality (MCU 2004)* (M. Margenstern, ed.), Saint Petersburg, Russia, 2004, Berlin, Heidelberg: Springer, 2004 pp. 188–199. doi:10.1007/978-3-540-31834-7_15.

[21] E. R. Berlekamp, J. H. Conway and R. K. Guy, *Winning Ways for Your Mathematical Plays,* 2nd ed., Natick, MA: A. K. Peters, 2004.

[22] G. J. Martínez, A. Adamatzky and H. V. McIntosh, "A Computation in a Cellular Automaton Collider Rule 110," *Advances in Unconventional Computing* (A. Adamatzky, ed.), Cham, Switzerland: Springer, 2017 pp. 391–428. doi:10.1007/978-3-319-33924-5_15.

[23] H. Zenil, N. A. Kiani and J. Tegnér, "Algorithmic Information Dynamics of Persistent Patterns and Colliding Particles in the Game of Life." arxiv.org/abs/1802.07181.

[24] J. Secretan, N. Beato, D. B. D Ambrosio, A. Rodriguez, A. Campbell and K. O. Stanley, "Picbreeder: Evolving Pictures Collaboratively Online," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*, Florence, Italy, 2008, New York: ACM, 2008 pp. 1759–1768. doi:10.1145/1357054.1357328.

[25] J. Simon, "Ganbreeder." (Aug 30, 2019) github.com/joel-simon/ganbreeder.

[26] D. Ha, "Reinforcement Learning for Improving Agent Design." arxiv.org/abs/1810.03779.

[27] K. O. Stanley, "Compositional Pattern Producing Networks: A Novel Abstraction of Development," *Genetic Programming and Evolvable Machines*, 8(2), 2007 pp. 131–162. doi:10.1007/s10710-007-9028-8.

[28] J. Collins, W. Geles, D. Howard and F. Maire, "Towards the Targeted Environment-Specific Evolution of Robot Components," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '18)*, Kyoto, Japan, 2018, New York: ACM, 2018 pp. 61–68. doi:10.1145/3205455.3205541.

[29] K. O. Stanley and R. Miikkulainen, "Evolving Neural Networks through Augmenting Topologies," *Evolutionary Computation*, **10**(2), 2002 pp. 99–127. doi:10.1162/106365602320169811.

[30] J. Lehman and K. O. Stanley, "Abandoning Objectives: Evolution through the Search for Novelty Alone," *Evolutionary Computation*, **19**(2), 2011 pp. 189–223. doi:10.1162/EVCO_a_00025.

[31] J. K. Pugh, L. B. Soros, and K. O. Stanley, "Quality Diversity: A New Frontier for Evolutionary Computation," *Frontiers in Robotics and AI*, **3**, 2016 40. doi:10.3389/frobt.2016.00040.

[32] A. Baranes and P.-Y. Oudeyer, "Active Learning of Inverse Models with Intrinsically Motivated Goal Exploration in Robots," *Robotics and Autonomous Systems*, **61**(1), 2013 pp. 49–73. doi:10.1016/j.robot.2012.05.008.

[33] H. Zenil, N. A. Kiani, and J. Tegnér, "Symmetry and Correspondence of Algorithmic Complexity over Geometric, Spatial and Topological Representations," *Entropy*, **20**(7), 2018 534. doi:10.3390/e20070534.

[34] S. Vaughan, "Random Time Series in Astronomy," *Philosophical Transactions of the Royal Society A: Mathematical, Physical & Engineering Sciences*, **371**(1984), 2013 20110549. doi:10.1098/rsta.2011.0549.

[35] C. Bays, "Candidates for the Game of Life in Three Dimensions," *Complex Systems*, **1**(3), 1987 pp. 373–400. complex-systems.com/pdf/01-3-1.pdf.

[36] K. Imai, Y. Masamori, C. Iwamoto and K. Morita, "On Designing Gliders in Three-Dimensional Larger than Life Cellular Automata," in *Natural Computing, Proceedings in Information and Communications Technology, Vol. 2* (F. Peper, H. Umeo, N. Matsui and T. Isokawa, eds.), Tokyo: Springer, 2010 pp. 184–190. doi:10.1007/978-4-431-53868-4_21.

[37] A. Adamatzky, A. Wuensche and B. D. L. Costello, "Glider-Based Computing in Reaction-Diffusion Hexagonal Cellular Automata," *Chaos, Solitons & Fractals*, **27**(2), 2006 pp. 287–295. doi:10.1016/j.chaos.2005.03.048.

[38] A. P. Goucher, "Gliders in Cellular Automata on Penrose Tilings," *Journal of Cellular Automata*, **7**(5–6), 2012 pp. 385–392.

[39] B. Bochenek and K. Tajs-Zielińska, "GOTICA - Generation of Optimal Topologies by Irregular Cellular Automata," *Structural and Multidisciplinary Optimization*, **55**(6), 2017 pp. 1989–2001. doi:10.1007/s00158-016-1614-z.

[40] N. Fatès, "A Guided Tour of Asynchronous Cellular Automata," in *Cellular Automata and Discrete Complex Systems (AUTOMATA 2013)* (J. Kari, M. Kutrib and A. Malcher, eds.), Berlin, Heidelberg: Springer, 2013 pp. 15–30. doi:10.1007/978-3-642-40867-0_2.

[41] C. Ryan, J. Fitzgerald, T. Kowaliw, R. Doursat, S. Carrignon and D. Medernach, "Evolution of Heterogeneous Cellular Automata in Fluctuating Environments," in *ALIFE 2016: The Fifteenth International Conference on the Synthesis and Simulation of Living Systems*, Cancún, Mexico, 2016, Cambridge, MA: The MIT Press, 2016 pp. 216–223. doi:10.7551/978-0-262-33936-0-ch041.

[42] P.-Y. Louis and F. R. Nardi, eds., *Probabilistic Cellular Automata: Theory, Applications and Future Perspectives*, Switzerland: Springer International Publishing, 2018.

[43] M. A. Bedau, "Artificial Life: Organization, Adaptation and Complexity from the Bottom Up," *Trends in Cognitive Sciences*, **7**(11), 2003 pp. 505–512. doi:10.1016/j.tics.2003.09.012.

[44] S. A. Benner, "Defining Life," *Astrobiology*, **10**(10), 2010 pp. 1021–1030. doi:10.1089/ast.2010.0524.

[45] P. Forterre, "Defining Life: The Virus Viewpoint," *Origins of Life and Evolution of the Biosphere*, **40**(2), 2010 pp. 151–160. doi:10.1007%2 Fs11084-010-9194-1.

[46] T. Taylor, M. Bedau, A. Channon, D. Ackley, W. Banzhaf, G. Beslon, E. Dolson, et al., "Open-Ended Evolution: Perspectives from the OEE Workshop in York," *Artificial Life*, **22**(3), 2016 pp. 408–423. doi:10.1162/ARTL_a_00210.