Turing-like Patterns from Cellular Automata

Gary R. Greenfield
Department of Mathematics & Computer Science
University of Richmond
Richmond, VA 23173, USA
ggreenfi@richmond.edu

Abstract

Turing patterns are patterns produced using a model proposed by Turing for vertebrate skin patterns. Turing-like patterns are similar patterns that are produced using models that differ from the one Turing proposed. We survey mathematical art based on Turing-like patterns and then focus on a cellular automaton for Turing-like patterns proposed by Young. By investigating the parameter space of Young's model and considering variations of his model we are led to a series of artworks based on Turing-like patterns.

Introduction

Turing patterns and Turing-like patterns trace to a proposal in 1952 by Alan Turing for modeling stable spatial patterns using systems where two or more chemicals, or morphogens, react while diffusing through a substrate at differing rates [8]. For biological modeling applications, the idea is that one morphogen determines short-range positive feedback and the other long-range inhibition (see, for example, Meinhardt [6] or Kondo and Miura [3]). Turing's hope was that his morphogenesis model would provide a biological *explanation* for vertebrate skin patterns. Although Turing's model for vertebrate skin pattern formation has been superseded [7], it continues to have an influence on activator-inhibitor and reaction-diffusion modeling and has been shown to occur in chemical systems [1].

The earliest model we are aware of that feasibly implements a simulation of Turing's activation-inhibition concept is a discrete cellular automaton model proposed by Young [11]. In 1993, in a collaboration with artist Sarah Stengle, I generated a series of Turing patterns using carefully selected parameter settings for Young's automata [2] to generate source material for drawings (Figure 1) and 3d-works (Figure 2) subsequently executed by Stengle [9, pp. 34–35].

In 2010, McCabe experimented with a simpler model than Young's, but one that was still based on Turing's activator-inhibitor concept, using only one substance representing the "pigmentation" level for each cell in an array in order to produce Turing-like patterns [5]. The description of his method is worth quoting:

> Diffusion is simulated by taking an average of the values of a certain area around each element, a smaller area for the activator and a larger area for the inhibitor. A simple rule is iteratively applied to all of the array elements for (usually) hundreds of time steps: If the average concentration in the smaller area is greater than the average concentration in the larger area, increase the value by a small amount, otherwise decrease the value by a small amount. At each time step the resulting values are re-normalised to the somewhat arbitrary range -1.0 to 1.0, to avoid "run away". Periodic boundary conditions are imposed to avoid edge effects.

By inducing [cyclic] symmetry, aggregating compound instances of the simple model, and varying the sizes of the areas used in his update rule McCabe generated a series of mathematical artworks which he called "multi-scale" Turing patterns.

In stark contrast to McCabe, in 2015 Werth described generating graphic designs that resemble Turing patterns — and led to abstract acrylic paintings — that were formed by repeatedly blurring and sharpening

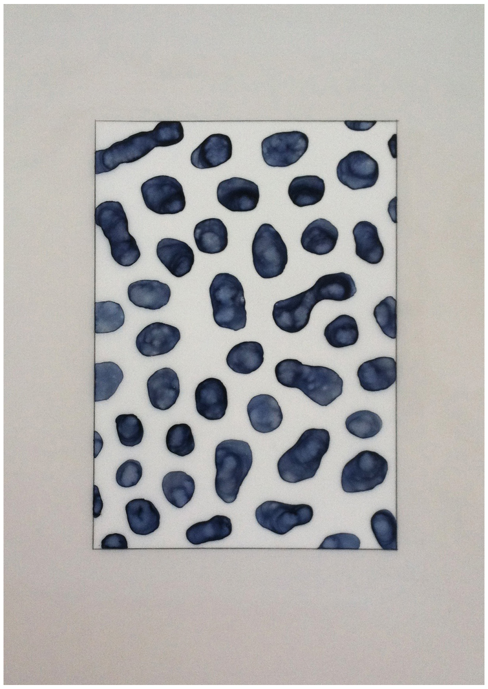

Figure 1: *"Greenfield Drawing", Sarah Stengle, 1994. Ink on transparent vellum, sheet size* $11'' \times 14''$*, image size* $7'' \times 10''$*.*

a seed image using Adobe Photoshop [10]. Werth provides an argument for why his "unsharp masking" technique "solves" the system of reaction-diffusion partial differential equations found in, say, Maini et al. [4] that are often used as a basis for simulating Turing-like patterns.

Because we feel it is the richer model, in this paper we return to Young's cellular automaton model to investigate more fully the parameters it uses and to introduce some new ideas for making artworks from the patterns it generates.

Young's Model

The cellular automaton Young proposes as a model for vertebrate skin patterns is a *discretized* solution to a generalized diffusion equation. The model assumes there is an embryonic skin consisting of uniformly sized cells. Some of the cells are differentiated, or colored, cells (DC's) while the remainder are undifferentiated cells (UC's). The DC's produce both an activator and an inhibitor. The activator stimulates nearby UC's to become DC's and the inhibitor discourages faraway DC's from remaining so. The activator morphogen's effect on cells a distance R away from a DC is $w_1(R)$ where $w_1(r) > 0$ while the inhibitor morphogen's effect at the same distance is $w_2(R)$ where $w_2(R) < 0$. Together they combine to induce a morphogenic field which has positive field value in a small circular region about each DC decreasing continuously until it changes to negative values in an annulus surrounding the circular region. We let r_1 and r_2 where $0 < r_1 < r_2$ be the inner and outer radii defining this annulus.

To describe the cellular automaton Young makes a DC's nearby activation effect constant at the *positive* value w_1, and annular inhibitor effect constant at the *negative* value w_2. Now, at each time-step, for each cell, the total effect of activator-inhibitor can be summed (it is under the influence of all DC's within the fixed radius r_2), so that it becomes, or remains, a DC if it has positive morphogen concentration and becomes, or remains, a UC otherwise.

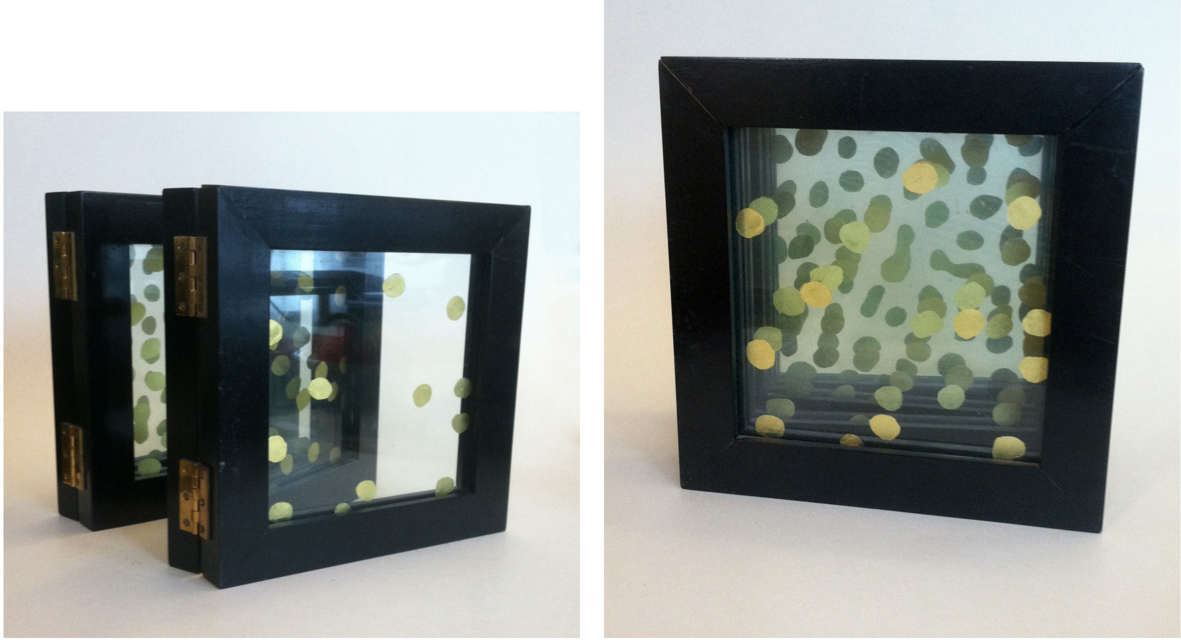

Figure 2 : *Two views of "Reaction Diffusion Glass Book", 1994. Gold leaf on glass in a painted wooden frame with brass hinges,* $10'' \times 10'' \times 8''$. *Reaction diffusion modeled by Gary Greenfield, book by Sarah Stengle.*

The key implementation decision Young makes is to normalize w_1 to $+1.0$ so that only w_2 varies. Starting with a fixed percentage of pseudo-randomly distributed DC's always produces the same general pattern with convergence resulting after approximately five iterations. Young's other assumption is that the activator-inhibitor field will sum to zero which means $w_1 r_1 + w_2(r_2 - r_1)$ should be approximately zero. We will not be constrained by this latter criterion.

For efficiency our implementation calculates the morphogenic effect for each cell using a three-pass procedure. An initial pass zeros out all cells' w values. In the second pass, for every DC encountered, a rectangular neighborhood large enough to include all cells within distance r_2 is traversed. For each cell within the neighborhood, its distance is calculated to the DC so that a decision can be made on how to update the total morphogen or w value of that cell. That is, if the cell is within r_1 units of the DC under consideration +1.0, the w_1 value, is added to its w value, but if it is between r_1 and r_2 units away from the DC under consideration w_2 is added to its w value. Note that the latter actually decreases the w value because w_2 is negative. A final pass then assigns cells to be DC's or UC's according to the accumulated w value. Note that if no cells change state during the final pass our algorithm detects that an equilibrium state for the pattern has been reached. We use periodic boundary conditions which implies the patterns will seamlessly tile the plane.

Exploring the Parameter Space

Assuming the initialization for the desired percentage of DC's is random, Young's cellular automaton model is controlled by only four parameters: the probability p a cell is initially assigned to be a DC, the morphogen inhibitor value w_2, the inner annulus radius r_1 and the outer annulus radius r_2. To explore this parameter space we conducted experiments where we fixed three of the parameters and assigned incremental values for the fourth. We ran the automaton until it reached equilibrium at each setting, and were careful to use the same seed for the pseudo-random number generator so that the DC initialization pattern would be the same for each run.

We let the default settings be $p = 0.16$, $r_1 = 5.01$, $r_2 = 15.01$, and $w_2 = -0.275$. The grid resolution was 200×200 and the cell resolution was 3×3, so the image resolution was 600×600 pixels. Varying p from 0.0001 to 0.001 by increments of 0.0001, from 0.001 to 0.01 by increments of 0.001, and from 0.01 to 0.3 by increments of 0.01 yielded the representative sequence of initial and final equilibrium examples shown in Figure 3. Varying r_1 from 4.51 to 8.51 by increments of 0.2 yielded the representative examples shown in Figure 4. Varying r_2 from 9.01 to 18.01 by increments of 0.5 yielded the representative examples shown in Figure 5. Finally, varying w_2 from -0.0851 to -0.3051 by increments of -0.005 yielded the representative examples shown in Figure 6.

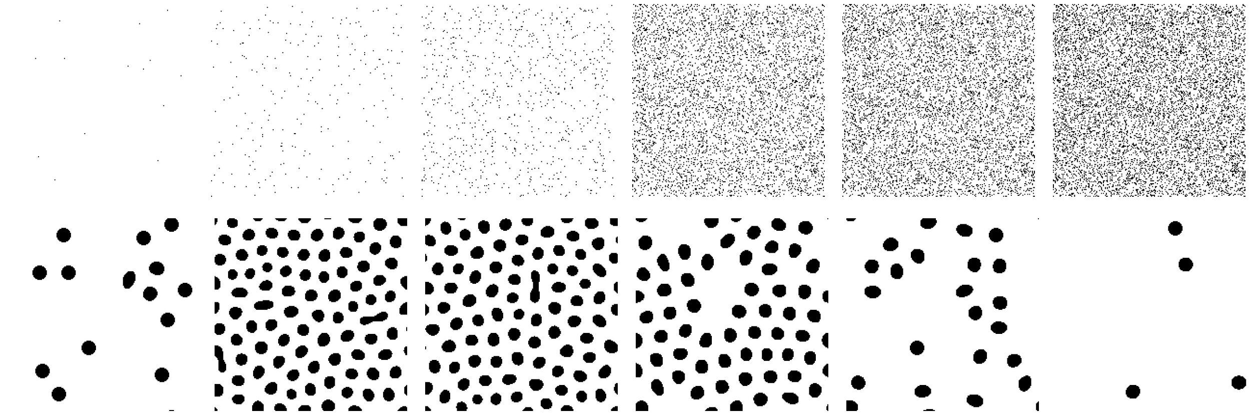

Figure 3: *Turing-like patterns. The top row shows the initial pattern when the probability p of being a differentiated cell is 0.0004, 0.006, 0.02, 0.13, 0.16 and 0.2 respectively, while the bottom row shows the corresponding equilibrium configuration reached after 30, 106, 98, 158, 44, and 4 iterations respectively.*

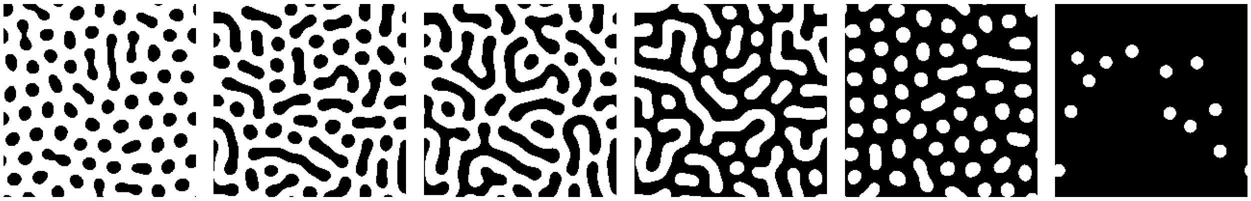

Figure 4: *Turing-like patterns when the inner radius r_1 is 5.71, 6.51, 6.91, 7.11, 7.51 and 7.91 yielded the equilibrium configurations shown after 94, 48, 67, 107, 145 and 27 iterations respectively.*

Patterns with a Gradient Parameter

We now consider Turing-like patterns where one of the cell parameters varies according to a cell's distance from the left edge. Because our grids use periodic boundary conditions if we vary the parameter linearly from a minimum value to a maximum value, when equilibrium is reached a weird asymmetry to the pattern results. For this reason it is easier to gauge the effect of using a gradient if we ramp up to the maximum value at the middle of the grid and then ramp back down to the minimum value at the right edge. We use a 400×100 grid so the image resolution is 1200×300 pixels.

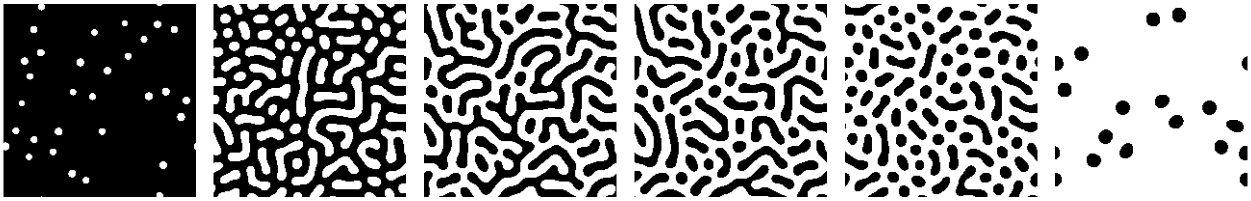

Figure 5: *Turing-like patterns when the outer radius r_2 is 10.01, 11.01, 11.51, 12.01, 12.51 and 15.01 yielded the equilibrium configurations shown after 6, 63, 46, 63, 47 and 22 iterations respectively.*

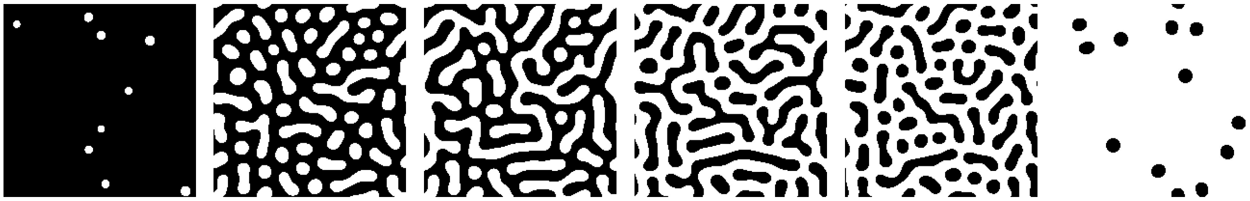

Figure 6 : *Turing-like patterns when the inhibitor factor w_2 is -0.0951, -0.1251, -0.1351, -0.1551, -0.1701 and -0.2851 yielded the equilibrium configurations shown after 9, 70, 42, 65, 57 and 19 iterations respectively.*

Figure 7 shows the initial and final configuration when p varies from 0.0015 to 0.22. Figure 8 shows examples of patterns obtained using the specified gradient ranges for each of the other three parameters.

Artworks Incorporating the Patterns

Following Werth, who colored the differentiated and undifferentiated cells, we applied color to the gradient patterns of the previous section. One modification we made was rather than ramping *linearly* between the minimum and maximum values of, in this case, w_2, we ramped using the square root of the normalized value (e.g., if we are ramping up, we first map x linearly to t in $[0, 1]$ and then map $\sqrt{t}$ linearly to $[-0.3, -0.1]$). We also ramped the color component values assigned to cells in a similar manner. With cell size decreased to 1×1 and grid dimensions increased to 1275×475, an example is shown in Figure 9.

As mentioned previously, by layering physical realizations of the patterns, in 1994 Sarah Stengle made 3d "glass books". Thanks to visual programming languages such as *Processing* which make it easier to implement transparency effects and perform image compositing than it was in 1993, we were able to digitally execute artworks that simulate this idea. We did not use a volumetric model. Instead, we seeded the psuedo-random number generator with the same value for each layer so that the same cells remained seed cells; we successively increased (or decreased) the w_2 value from layer to layer; we varied the color intensity and transparency from layer to layer; and we used image compositing to simulate stacking the layers. Examples using 2×2 cells, 300×300 grids and four layers are shown in Figure 10. As noted by Young, the automaton patterns tend to converge very fast. We found we obtained acceptable results by halting the automaton after fifteen iterations for each layer rather than letting it run until equilibrium was reached.

Young gave an example [11, Fig. 3] of a Turing-like pattern formed using concentric ellipses rather than concentric circles to define the inner and outer regions needed for his model. Changing w_2 to -0.1501 to provide further visual interest, in Figure 11 we exhibit a trio of patterns where the cells are 1×1, the grid is 600×600 and the inner and outer regions are concentric circles, diamonds and squares respectively. Note that this corresponds to using the Euclidean metric $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$, the taxicab metric $|x_2 - x_1| + |y_2 - y_1|$, or the supremum metric $\max(|x_2 - x_1|, |y_2 - y_1|)$ to measure distance between cells

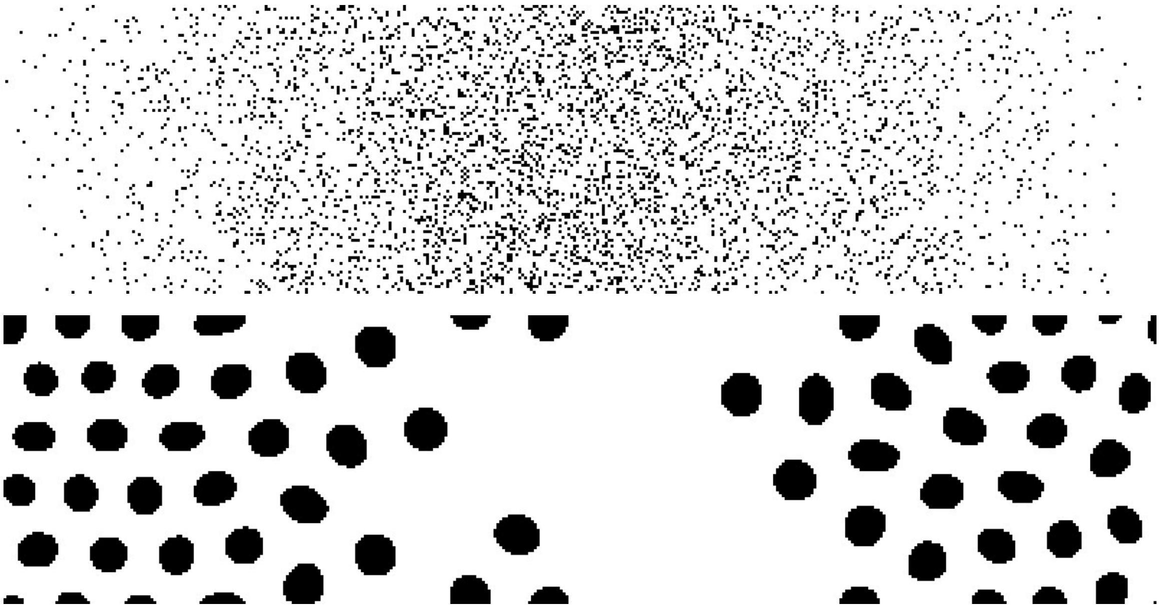

Figure 7 : *Initial pattern and final equilibrium pattern for a grid seeded so that the differentiated cell probability* p *ramps left to right from 0.0014 up to 0.22 and then back to 0.0014. Equilibrium was reached after 88 iterations.*

in the grid. Interestingly, the number of iterations to reach equilibrium was 79 for the circles and 105 for the diamonds, but soared to 948 for the squares. This discrepancy is due to the difference in areas across metrics between inner and outer regions under our default r_1 and r_2 settings.

Future Work

We believe that there is still much more that can be explored artistically using Young's cellular automata. For example, recently we began experimenting with using different w_2 values for different subregions of the grid. Since equilibrium patterns tile the plane, this idea can be used to create optical illusion effects.

References

[1] V. Castets, E. Dulos, J. Boissonade and P. De Kepper, Experimental evidence of a sustained standing Turing-type nonequilibrium chemical pattern, *Physical Review Letters*, vol. 64, 1990, pp. 2953–2956.

[2] Gary Greenfield, Case Study: A Sculptor - Programmer Collaboration, Technical Report TR-93-03, Mathematics and Computer Science, University of Richmond, 1993.

[3] S. Kondo and T. Miura, Reaction-diffusion model as a framework for understanding biological pattern formation, *Science*, 24 September 2010, pp. 1616–1620.

[4] P. K. Maini, K. J. Painter and H. N. P. Chau, Spatial pattern formation in chemical and biological systems, *Journal Chemical Society, Faraday Transactions*, vol. 93, no. 20, 1997, pp. 3601–3610.

[5] Jonathan McCabe, Cyclic symmetric multi-scale Turing patterns, in *Bridges 2010 Conference Proceedings*, George W. Hart and Reza Sarhangi, eds., Tessellations Publishing, Phoenix, AZ, 2015, pp. 387–390. Available online at `http://archive.bridgesmathart.org/2010/bridges2010-387.html`.

[6] H. Meinhardt, *Models of Biological Pattern Formation*, Academic Press, London, 1982.

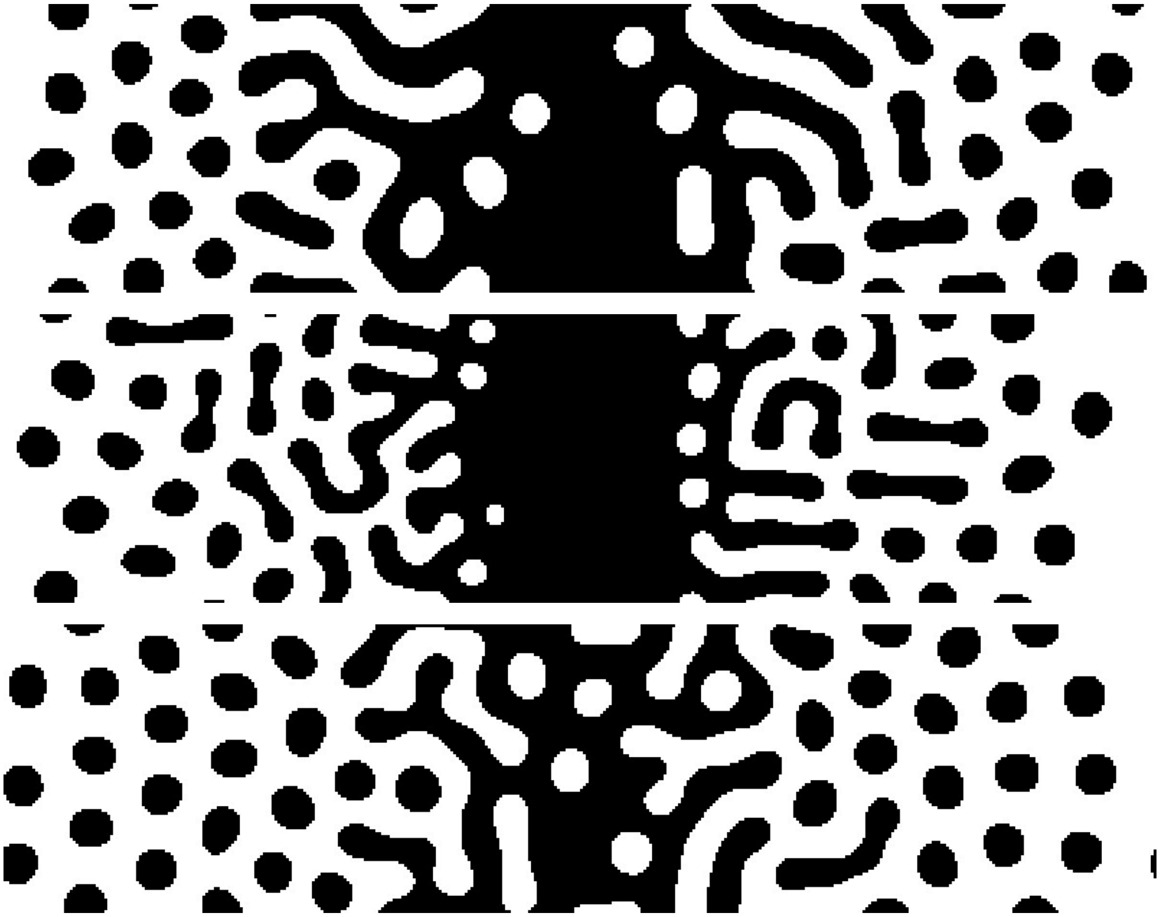

Figure 8: *Top to bottom: Final equilibrium configuration reached using gradient range 4.91 to 8.11 for the inner radius r_1, 9,01 to 16.51 for the outer radius r_2, and -0.3 to -0.1 for the inhibitor parameter w_2. Equilibrium was reached after 101, 172 and 105 iterations respectively.*

[7] J. D. Murray, Why are there no three-headed monsters? Mathematical modeling in biology, *Notices of the AMS*, vol. 59, no. 6, June/July 2012, pp. 785–795.

[8] A. Turing, The chemical basis of morphogenesis, *Philosophical Transactions of the Royal Society of London*, vol. B, no. 237, 1952, pp. 37–72.

[9] Sarah Stengle, *Without Me: Diverse Themes*, Exhibition catalog from Zoellner Art Center, Lehigh University, Bethlehem, PA, 2001.

[10] Andrew Werth, Turing patterns in Photoshop, in *Bridges 2015 Conference Proceedings*. Kelly Delp, Craig S. Kaplan, Douglas McKenna and Reza Sarhangi, eds., Tessellations Publishing, Phoenix, AZ, 2015, pp. 459–462. Available online at `http://archive.bridgesmathart.org/2015/bridges2015-459.html`.

[11] D. A. Young, A local activator-inhibitor model of vertebrate skin patterns, in *Theory & Application of Cellular Automata*, S. Wolfram (ed.), World Scientific, 1986, pp. 320–327.

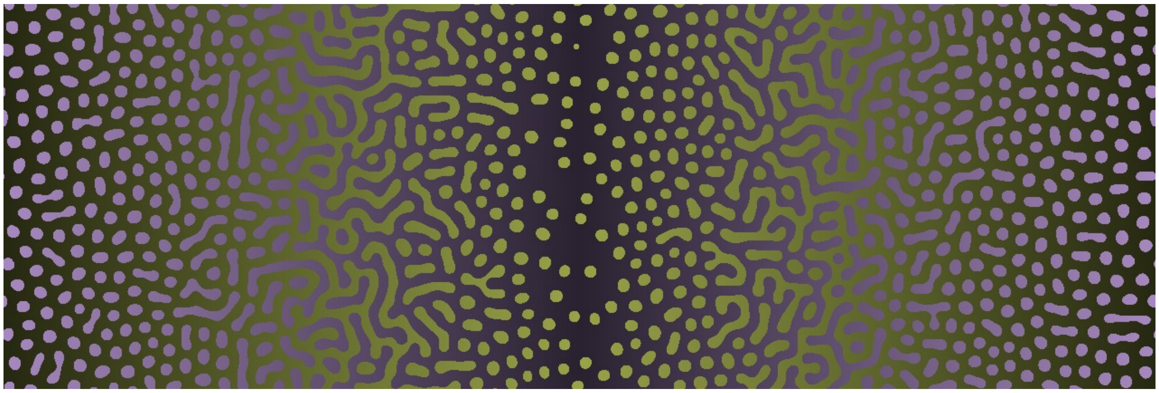

Figure 9 : *"Turing Gradient Pattern #30423", Gary Greenfield, 2015.*

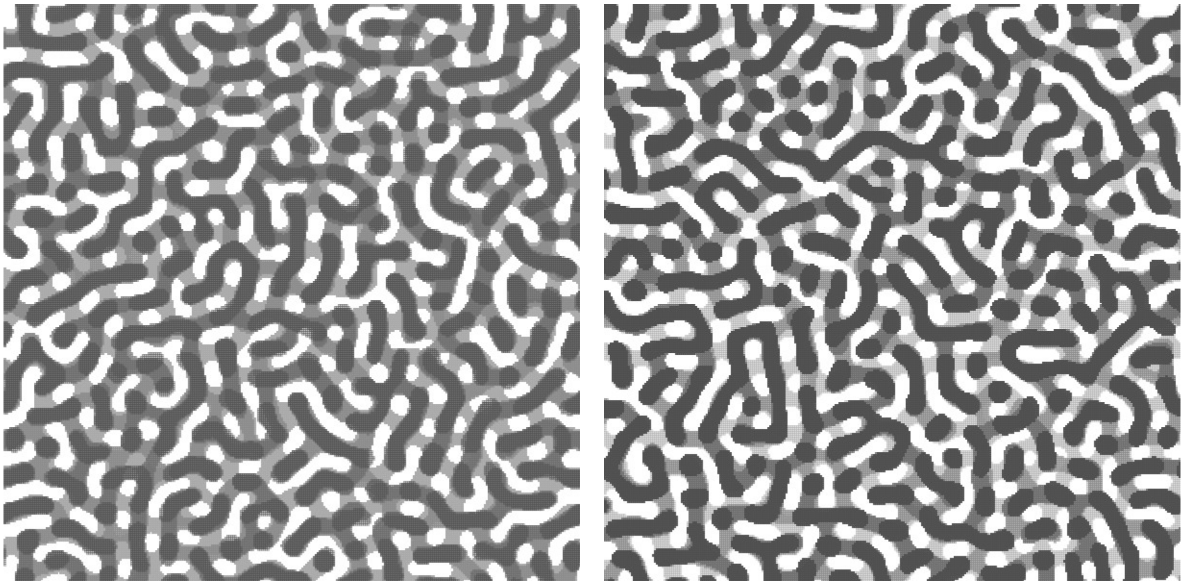

Figure 10 : *"Turing Transparency Pattern #47795" and "Turing Transparency Pattern #47884", Gary Greenfield, 2016.*

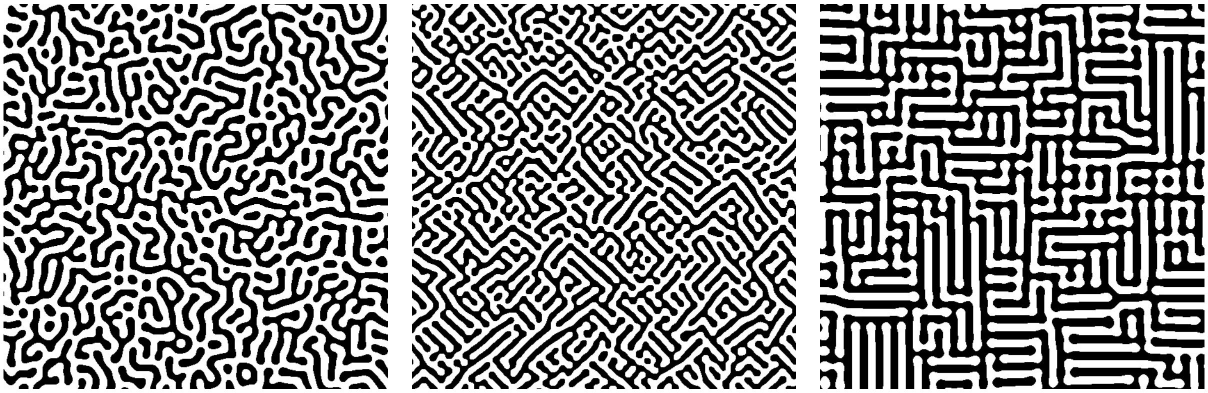

Figure 11 : *"Turing Metric Trio", Gary Greenfield, 2016.*