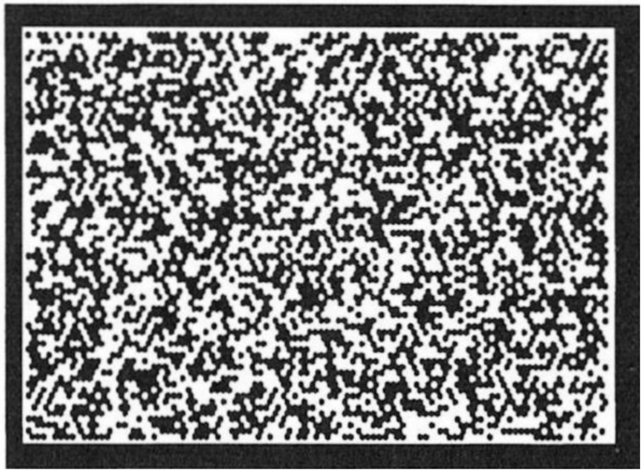
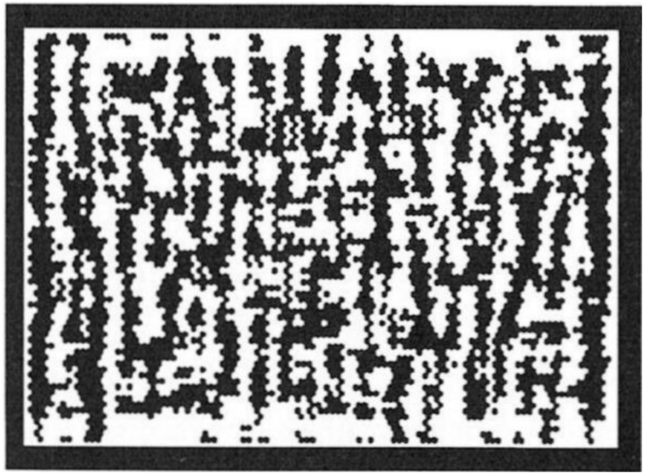
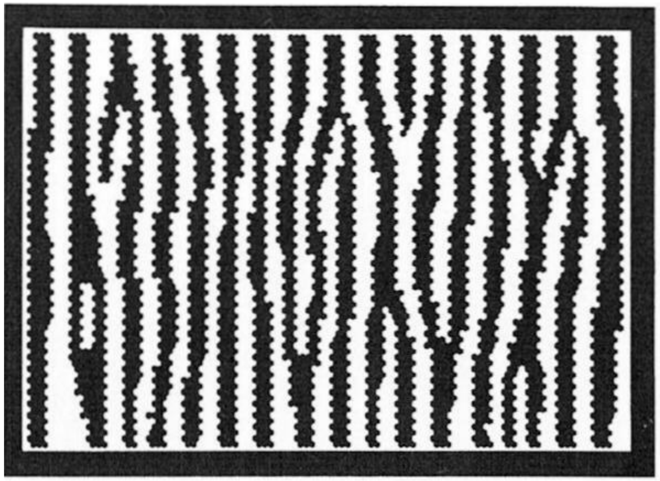
B.
Pattern Formation

Differentiation & Pattern Formation

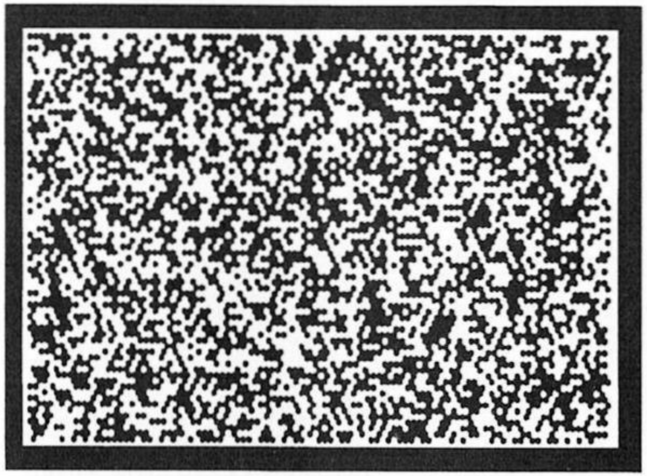
- A central problem in development: How do cells differentiate to fulfill different purposes?
- How do complex systems generate spatial & temporal structure?
- CAs are natural models of intercellular communication

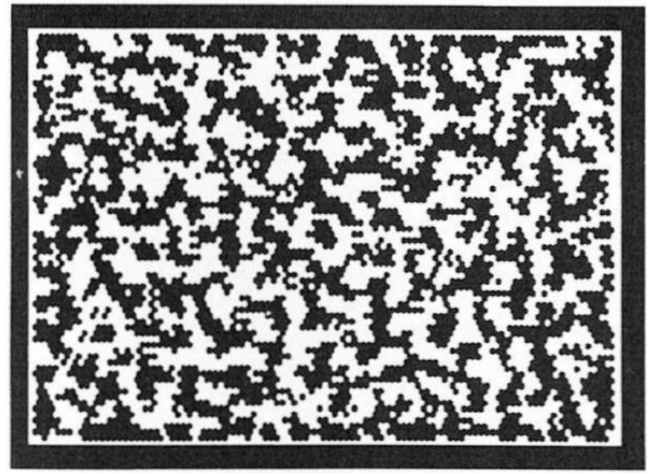
Zebra

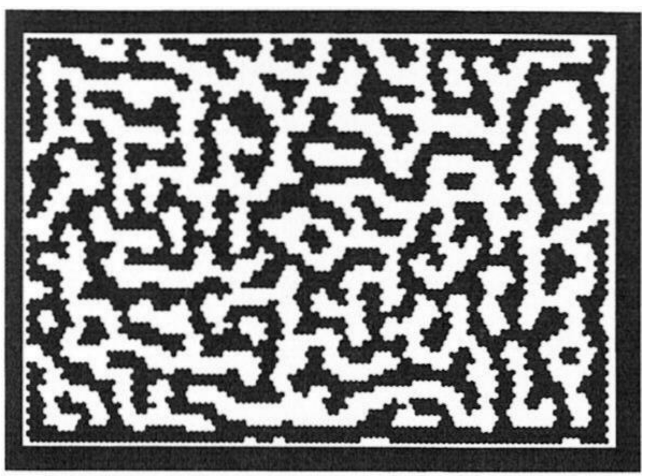




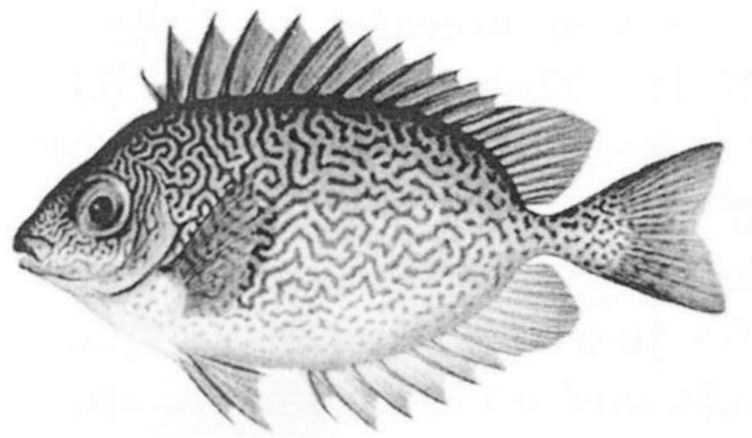

figs. from Camazine & al.: *Self-Org. Biol. Sys.*

Vermiculated Rabbit Fish







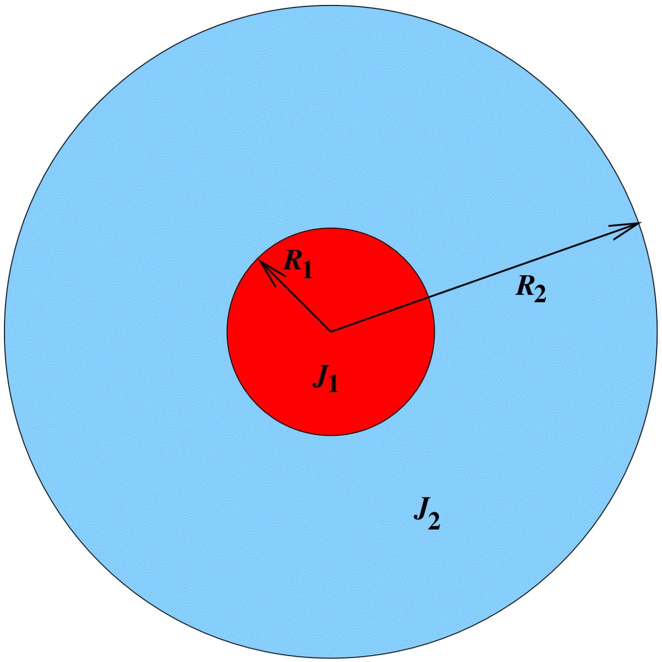


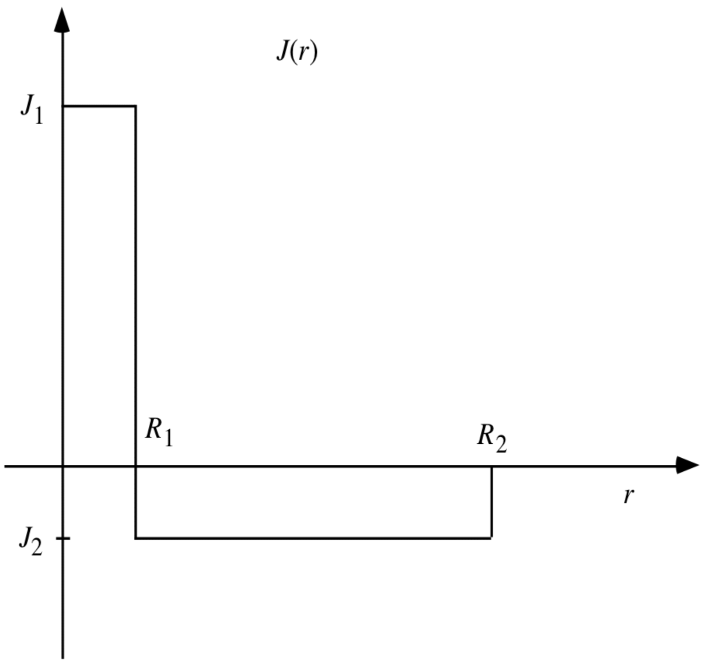
figs. from Camazine & al.: *Self-Org. Biol. Sys.*

Activation & Inhibition in Pattern Formation

- Color patterns typically have a characteristic length scale
- Independent of cell size and animal size
- Achieved by:
  - short-range activation $\Rightarrow$ local uniformity
  - long-range inhibition $\Rightarrow$ separation

Interaction Parameters





- R_1 and R_2 are the interaction ranges
- J_1 and J_2 are the interaction strengths

CA Activation/Inhibition Model

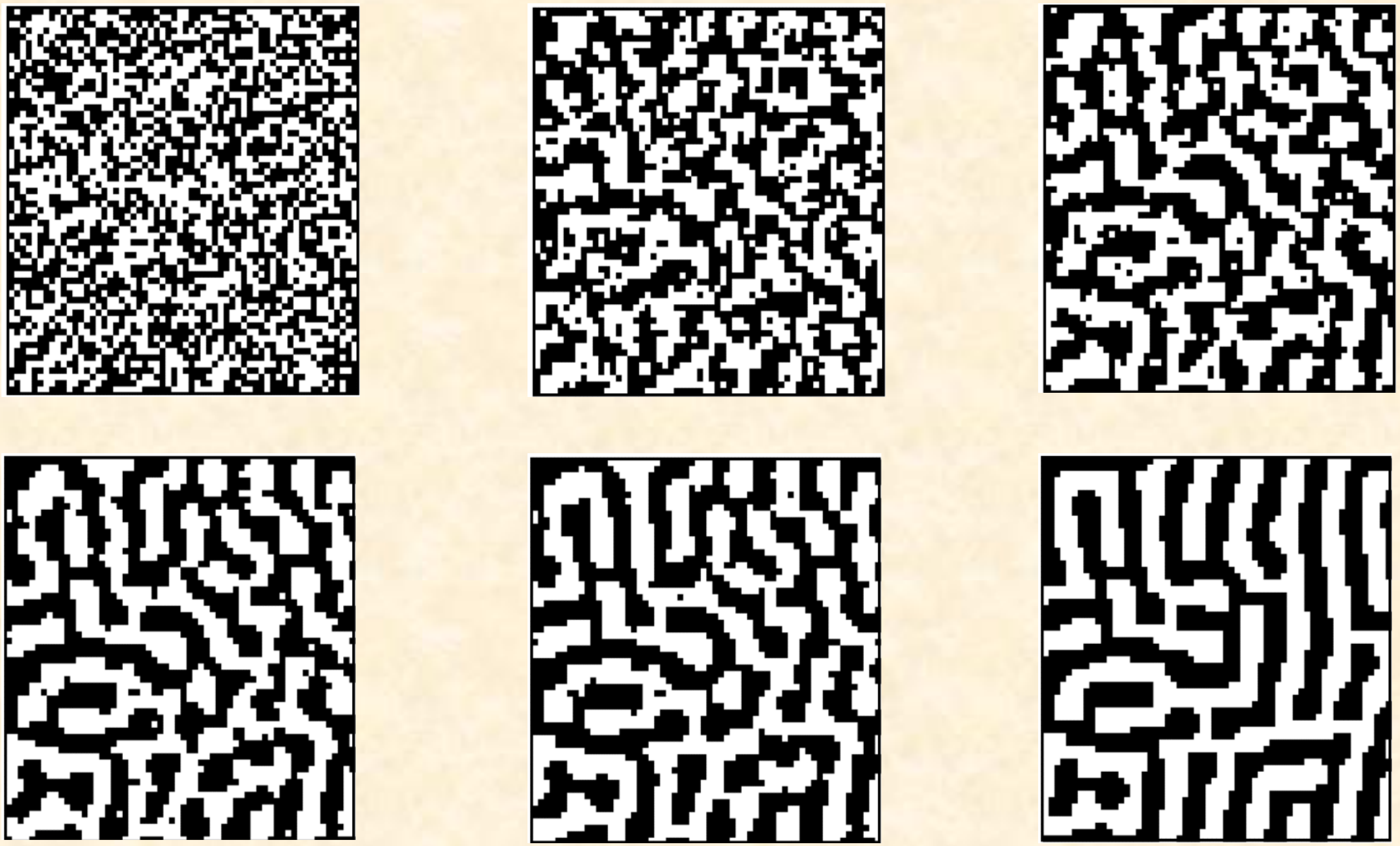
- Let states $s_i \in \{-1, +1\}$
- and h be a bias parameter
- and r_{ij} be the distance between cells i and j
- Then the state update rule is:

$$s_i(t+1) = \text{sign}\left[h + J_1 \sum_{r_{ij} < R_1} s_j(t) + J_2 \sum_{R_1 \le r_{ij} < R_2} s_j(t) \right]$$

Demonstration of NetLogo Program for Activation/Inhibition Pattern Formation: Fur

RunAICA.nlogo

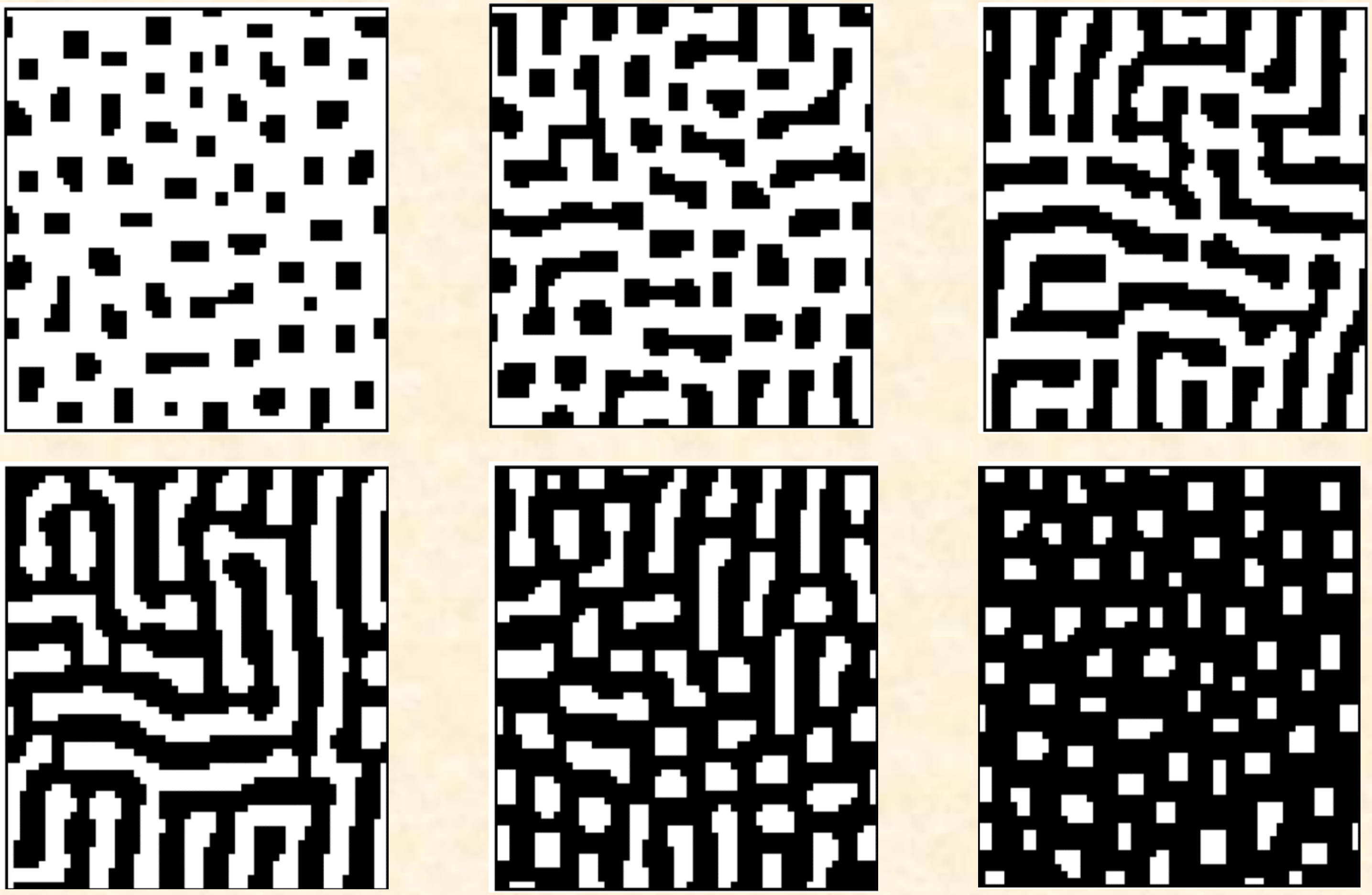
Example
(R_1=1, R_2=6, J_1=1, J_2=–0.1, h=0)



figs. from Bar-Yam

Effect of Bias

(h = –6, –3, –1; 1, 3, 6)



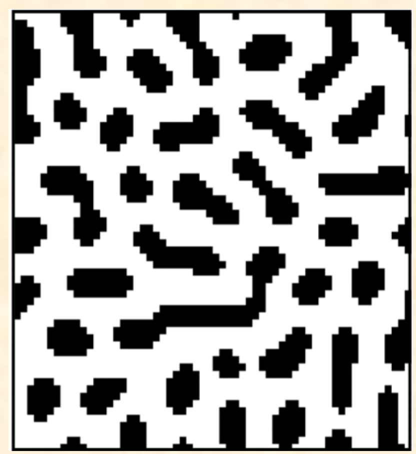
figs. from Bar-Yam

Effect of Interaction Ranges

$R_2 = 6$
$R_1 = 1$
$h = 0$

$R_2 = 8$
$R_1 = 1$
$h = 0$

$R_2 = 6$
$R_1 = 1.5$
$h = 0$



$R_2 = 6$
$R_1 = 1.5$
$h = -3$

figs. from Bar-Yam

Differential Interaction Ranges

- How can a system using strictly local interactions discriminate between states at long and short range?
- E.g. cells in developing organism
- Can use two different *morphogens* diffusing at two different rates
  - activator diffuses slowly (short range)
  - inhibitor diffuses rapidly (long range)

Digression on Diffusion

- Simple 2-D diffusion equation:

$$\dot{A}(x,y) = D\nabla^2 A(x,y)$$

- Recall the 2-D Laplacian:

$$\nabla^2 A(x,y) = \frac{\partial^2 A(x,y)}{\partial x^2} + \frac{\partial^2 A(x,y)}{\partial y^2}$$

- The Laplacian (like 2nd derivative) is:
  - positive in a local minimum
  - negative in a local maximum

Reaction-Diffusion System

$$\frac{\partial A}{\partial t} = D_{\mathrm{A}} \nabla^2 A + f_{\mathrm{A}}(A, I)$$

diffusion — reaction

$$\frac{\partial I}{\partial t} = D_{\mathrm{I}} \nabla^2 I + f_{\mathrm{I}}(A, I)$$

$$\frac{\partial}{\partial t}\begin{pmatrix} A \\ I \end{pmatrix} = \begin{pmatrix} D_{\mathrm{A}} & 0 \\ 0 & D_{\mathrm{I}} \end{pmatrix}\begin{pmatrix} \nabla^2 A \\ \nabla^2 I \end{pmatrix} + \begin{pmatrix} f_{\mathrm{A}}(A, I) \\ f_{\mathrm{I}}(A, I) \end{pmatrix}$$

$$\dot{\mathbf{c}} = \mathbf{D}\nabla^2\mathbf{c} + \mathbf{f}(\mathbf{c}), \quad \text{where } \mathbf{c} = \begin{pmatrix} A \\ I \end{pmatrix}$$

General Reaction-Diffusion System

$$\frac{\partial c_i}{\partial t} = \sum_{\alpha} k_\alpha \nu_{i\alpha} \left(\prod_{k=1}^{n} c_k^{m_{k\alpha}} \right) - \nabla \cdot \mathbf{j}_i$$

where $\mathbf{j}_i = \vec{\mu}_i c_i - \mathbf{div}\ \mathbf{D}_i c_i$ (flux)

where $k_\alpha =$ rate constant for reaction α

and $\nu_{i\alpha} =$ stoichiometric coefficient

and $m_{k\alpha} =$ a non-negative integer

and $\vec{\mu}_i =$ drift vector

and $\mathbf{D}_i =$ diffusivity matrix

where $\mathbf{div}\ \mathbf{D}c = \sum_j \mathbf{e}_j \sum_k D_{jk} \, {\partial c}/{\partial x_k}$

Framework for Complexity

- change = source terms + transport terms
- source terms = local coupling
  = interactions local to a small region
- transport terms = spatial coupling
  = interactions with contiguous regions
  = advection + diffusion
  - advection: non-dissipative, time-reversible
  - diffusion: dissipative, irreversible

Continuous-time Activator-Inhibitor System

- Activator A and inhibitor I may diffuse at different rates in x and y directions
- Cell becomes more active if activator + bias exceeds inhibitor
- Otherwise, less active

$$\frac{\partial A}{\partial t} = d_{\mathrm{Ax}} \frac{\partial^2 A}{\partial x^2} + d_{\mathrm{Ay}} \frac{\partial^2 A}{\partial y^2} + k_{\mathrm{A}} \left(A + B - I \right)(1 - A)$$

$$\frac{\partial I}{\partial t} = d_{\mathrm{Ix}} \frac{\partial^2 I}{\partial x^2} + d_{\mathrm{Iy}} \frac{\partial^2 I}{\partial y^2} + k_{\mathrm{I}} \left(A + B - I \right)(1 - I)$$

NetLogo Simulation of Reaction-Diffusion System

1. Diffuse activator in X and Y directions
2. Diffuse inhibitor in X and Y directions
3. Each patch performs:

stimulation = bias + activator – inhibitor + noise

if stimulation > 0 then

  set activator and inhibitor to 100

else

  set activator and inhibitor to 0

Demonstration of NetLogo Program for Activator/Inhibitor Pattern Formation

Run Pattern.nlogo

Demonstration of NetLogo Program for Activator/Inhibitor Pattern Formation with Continuous State Change

Run Activator-Inhibitor.nlogo

Turing Patterns

- Alan Turing studied the mathematics of reaction-diffusion systems
- Turing, A. (1952). The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society* **B 237**: 37–72.
- The resulting patterns are known as *Turing patterns*

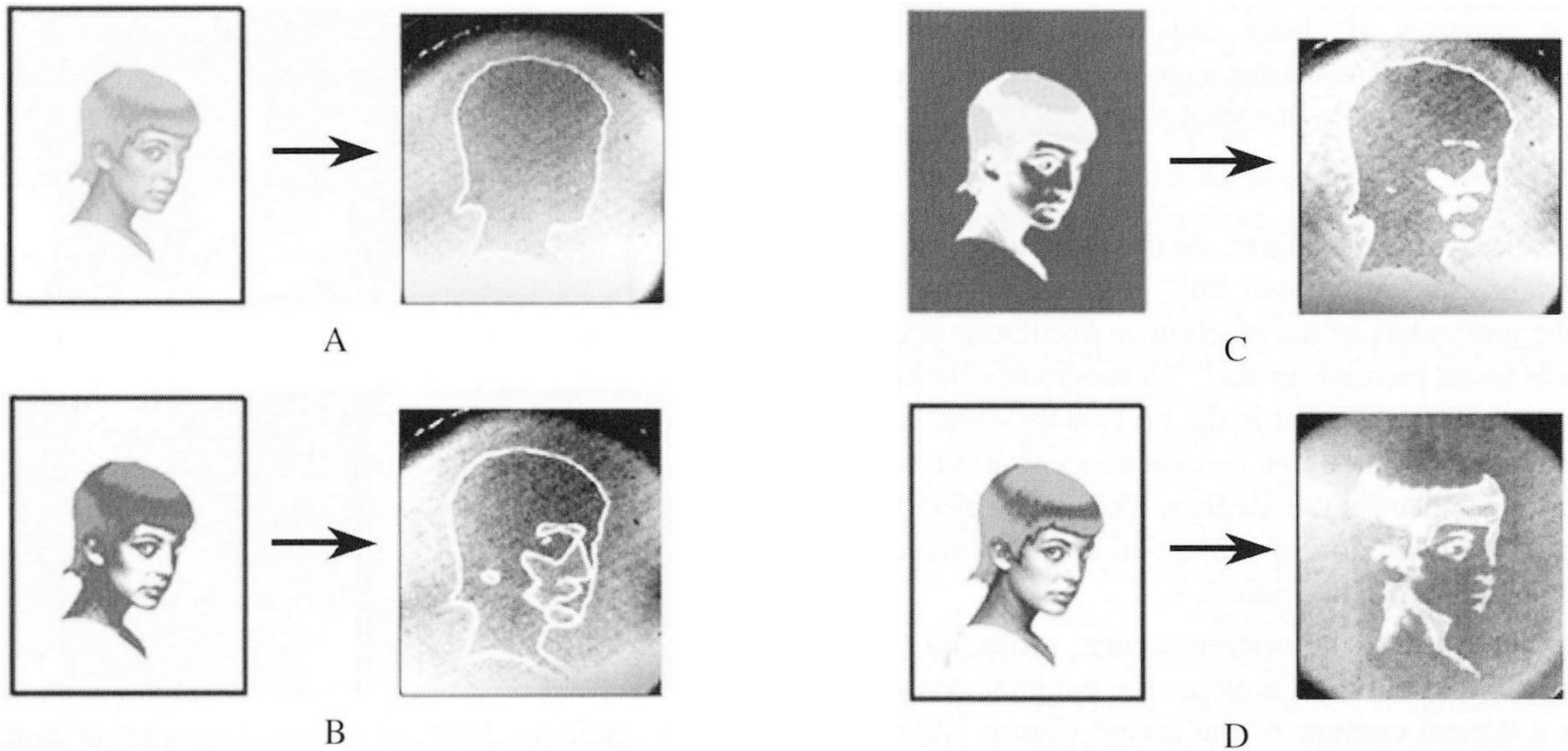
A Key Element of Self-Organization

- Activation vs. Inhibition
- Cooperation vs. Competition
- Amplification vs. Stabilization
- Growth vs. Limit
- Positive Feedback vs. Negative Feedback
  - Positive feedback creates
  - Negative feedback shapes

Reaction-Diffusion Computing

- Has been used for image processing
  - diffusion ⇒ noise filtering
  - reaction ⇒ contrast enhancement
- Depending on parameters, RD computing can:
  - restore broken contours
  - detect edges
  - improve contrast

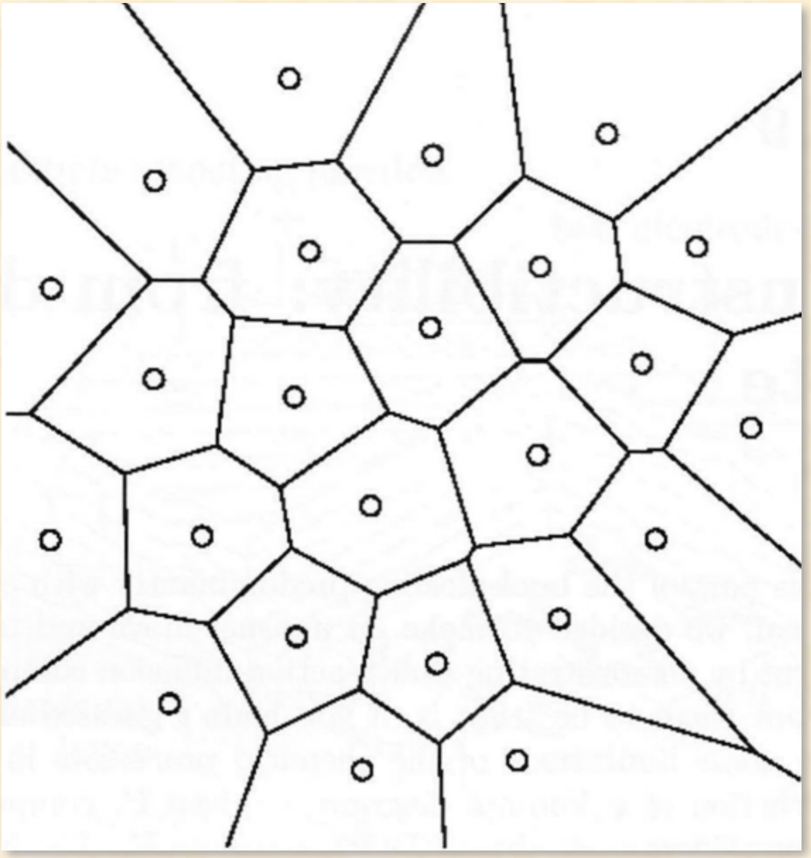
Image Processing in BZ Medium



- (A) boundary detection, (B) contour enhancement, (C) shape enhancement, (D) feature enhancement

Image < Adamatzky, *Comp. in Nonlinear Media & Autom. Coll.*

Voronoi Diagrams



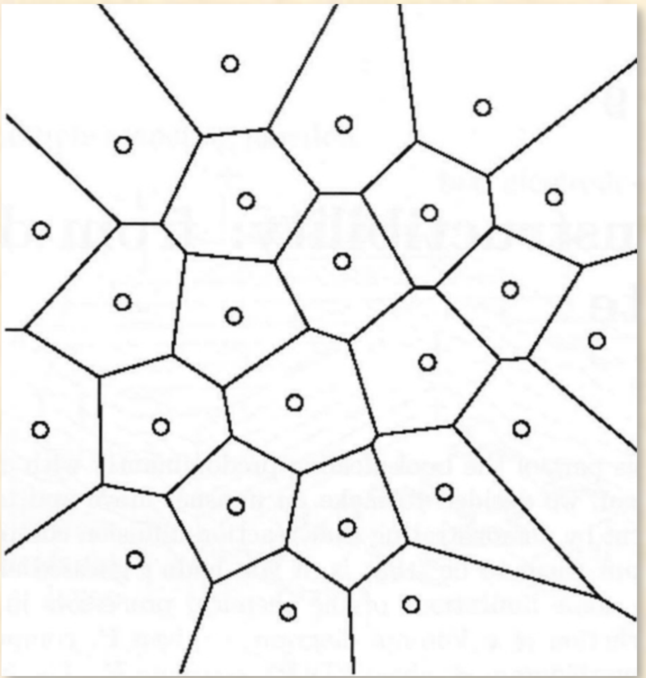
- Given a set of generating points:
- Construct a polygon around each generating point of set, so all points in a polygon are closer to its generating point than to any other generating points.

Image < Adamatzky & al., *Reaction-Diffusion Computers*

Some Uses of Voronoi Diagrams

- Collision-free path planning
- Determination of service areas for power substations
- Nearest-neighbor pattern classification
- Determination of largest empty figure

Computation of Voronoi Diagram by Reaction-Diffusion Processor



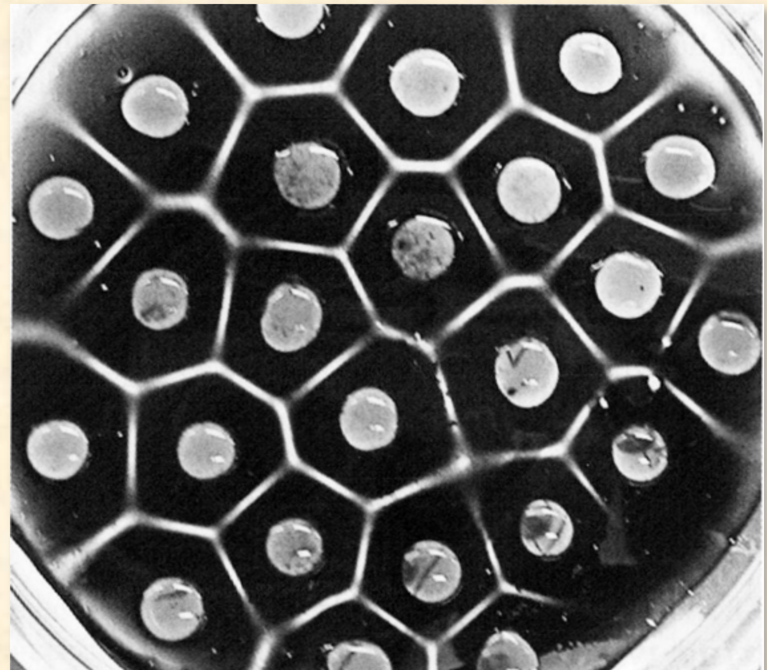


Image < Adamatzky & al., *Reaction-Diffusion Computers*

Mixed Cell Voronoi Diagram

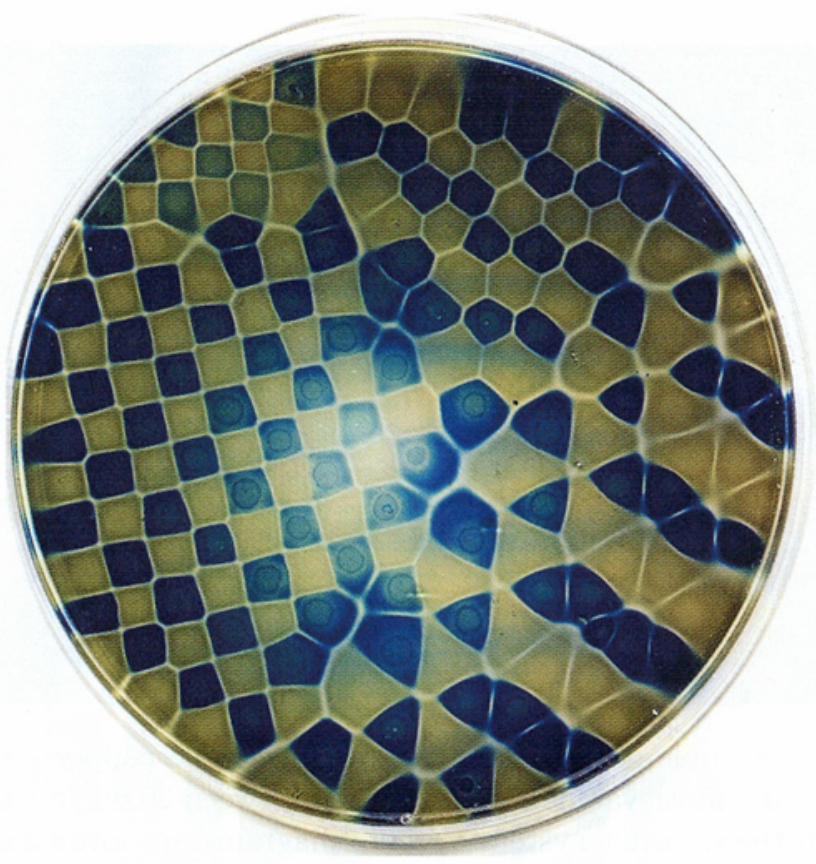


Image < Adamatzky & al., *Reaction-Diffusion Computers*

Path Planning via BZ medium: No Obstacles

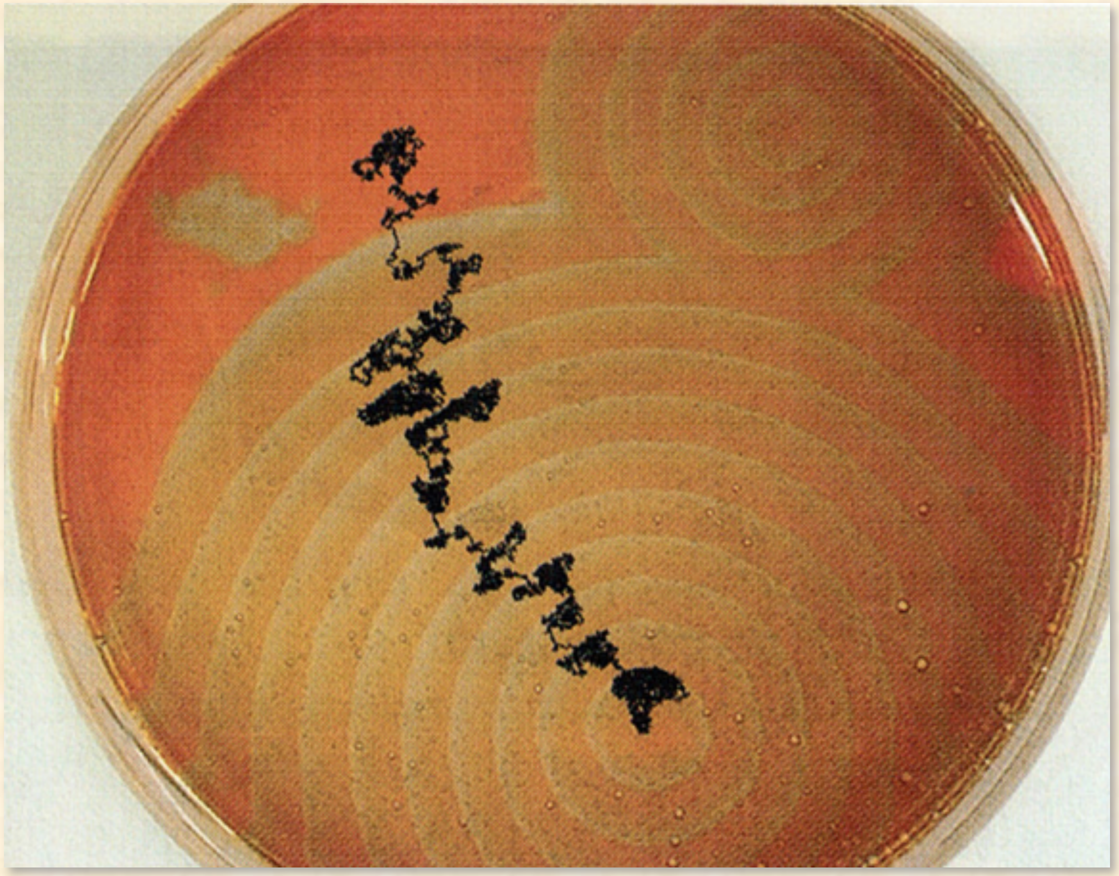


Image < Adamatzky & al., *Reaction-Diffusion Computers*

Path Planning via BZ medium: Circular Obstacles

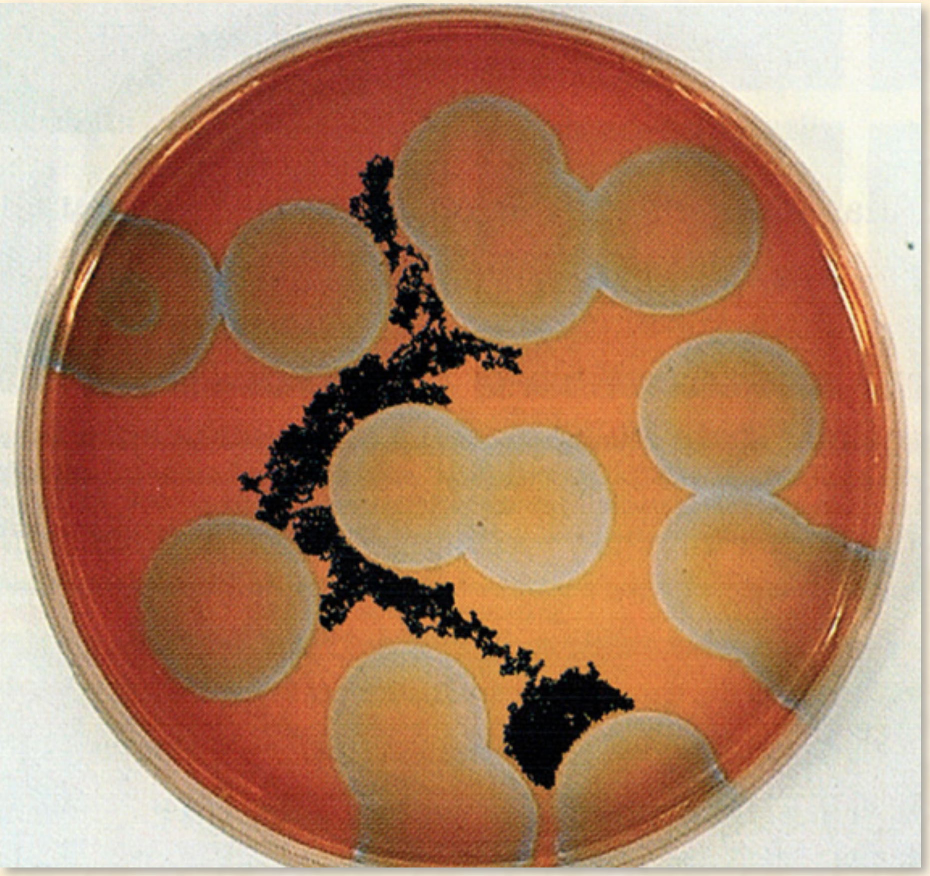


Image < Adamatzky & al., *Reaction-Diffusion Computers*

Mobile Robot with Onboard Chemical Reactor

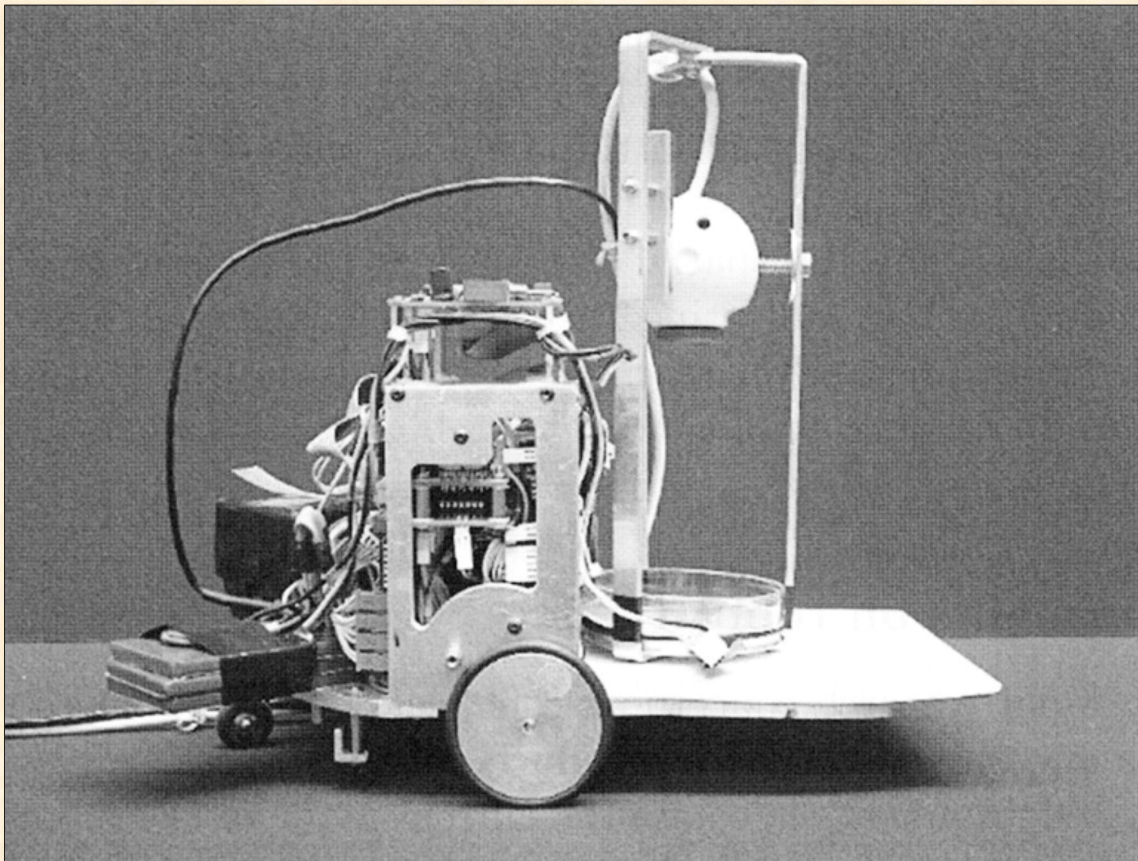


Image < Adamatzky & al., *Reaction-Diffusion Computers*

Actual Path: Pd Processor

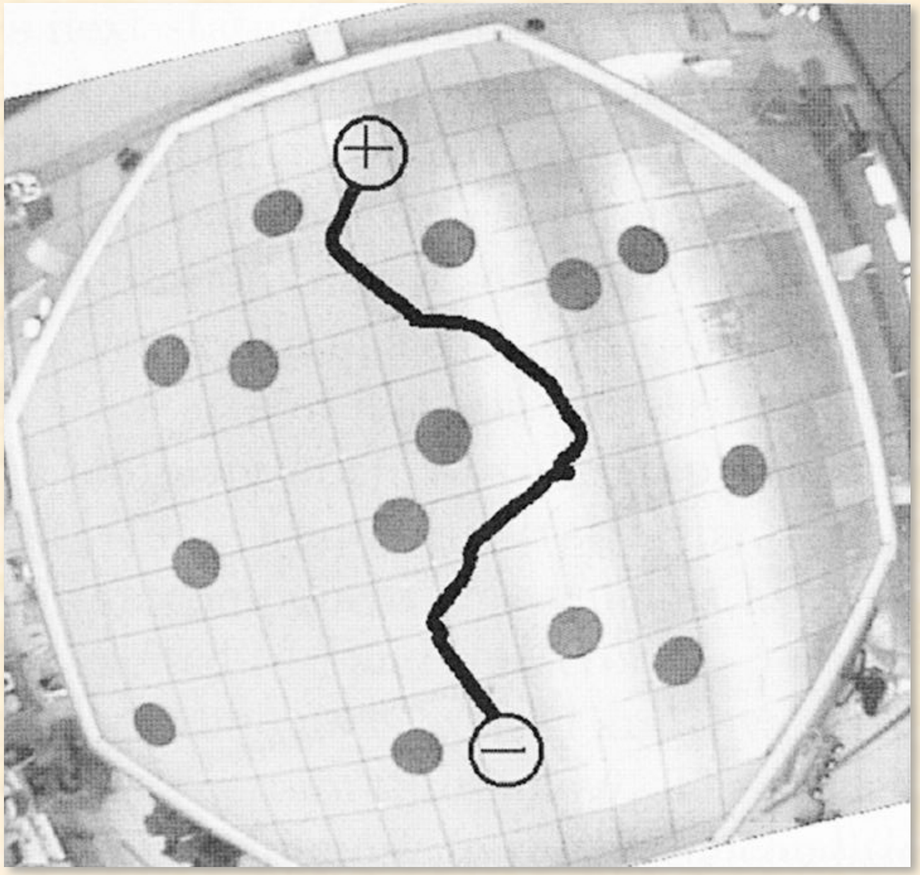


Image < Adamatzky & al., *Reaction-Diffusion Computers*

Actual Path: Pd Processor

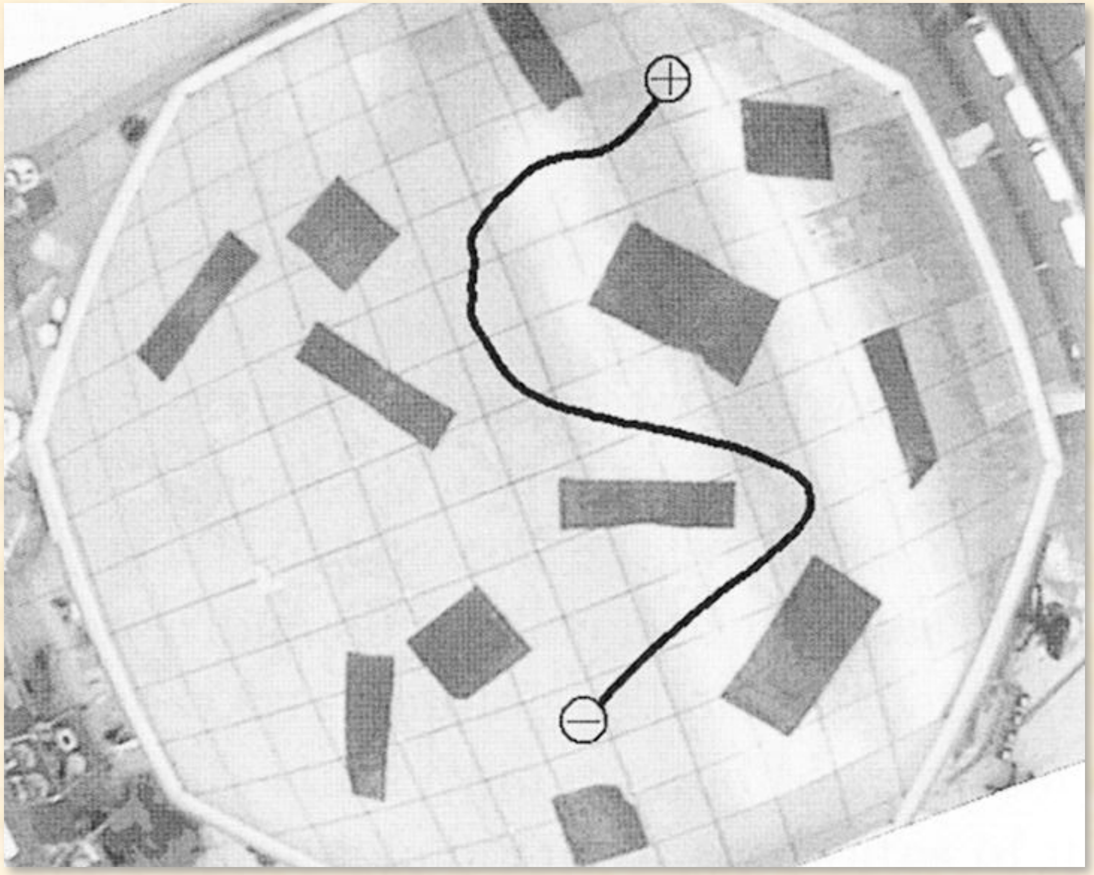


Image < Adamatzky & al., *Reaction-Diffusion Computers*

Actual Path: BZ Processor

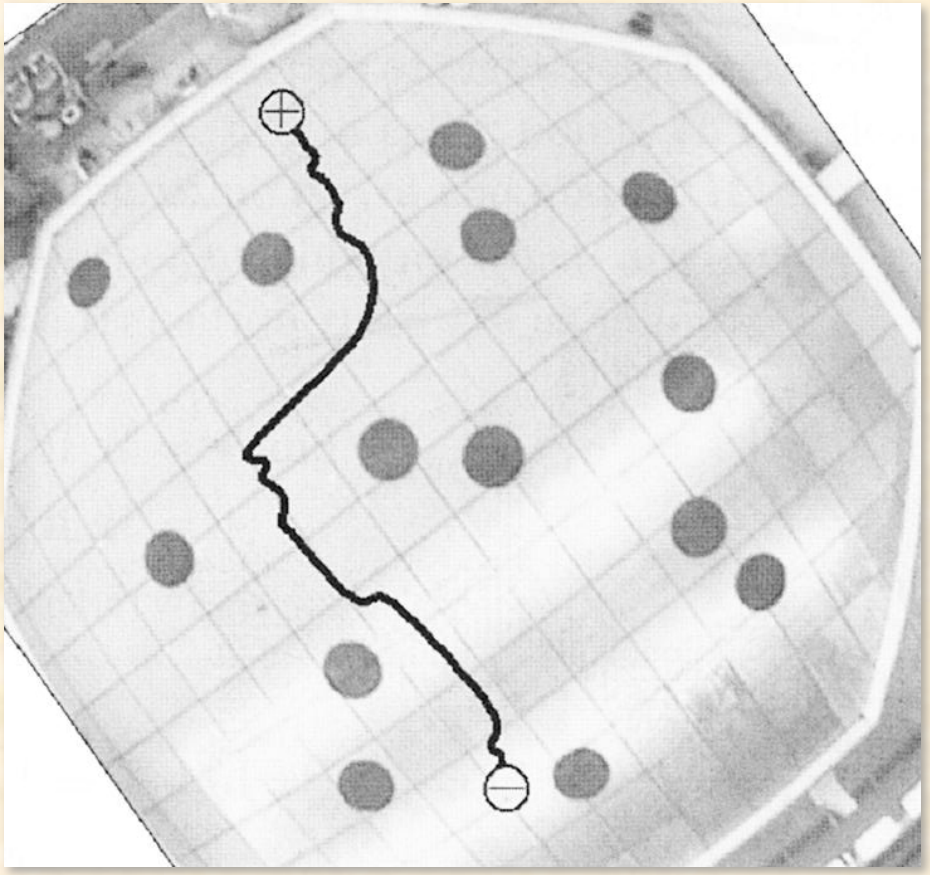


Image < Adamatzky & al., *Reaction-Diffusion Computers*

Bibliography for Reaction-Diffusion Computing

1. Adamatzky, Adam. *Computing in Nonlinear Media and Automata Collectives*. Bristol: Inst. of Physics Publ., 2001.
2. Adamatzky, Adam, De Lacy Costello, Ben, & Asai, Tetsuya. *Reaction Diffusion Computers*. Amsterdam: Elsevier, 2005.