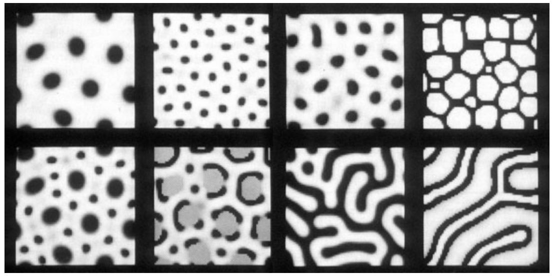
Reaction-Diffusion Textures Turk 91; Witkin & Kass 91

- Reaction-diffusion is a mathematical model for generation of natural patterns, arising due to local non-linear interactions of excitation and inhibition.
- First proposed by Alan Turing in 1952



Main Idea

- Certain cell properties (such as generation of pigments) are determined in the embryo based on the local concentration of one or more chemicals, called morphogens.
- Morphogen concentrations are determined by two concurrent processes:
  - Diffusion - spreading of morphogens through the tissue
  - Reaction - chemical reactions that create or destroy morphogens, based on their concentration in each cell
- Natural patterns can be generated by simulating the R-D processes.

Mathematical Model

- A system of non-linear partial differential equations.
- Let *C*(x,y) denote the concentration of a morphogen in a (2D) system. The governing equation is:

$$\dot{C} = \frac{\partial C}{\partial t} = a^2 \nabla^2 C - bC + R$$

- where a is the diffusion coefficient, b is the dissipation coefficient, R is the rate of change due to reaction, and $\nabla^2 C = \frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2}$ is the Laplacian of *C*.

Finite Differences Apprx

- On a 2D integer grid with spacing h, the Laplacian is approximated as:

$$\nabla^2 C \approx \frac{C_{i+1,j} + C_{i-1,j} + C_{i,j+1} + C_{i,j-1} - 4C_{i,j}}{h^2}$$

- In other words, the Laplacian operator is a convolution with:

$$L = \frac{1}{h^2}\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Finite Differences (cont'd)

The original equation $\dot{C} = \frac{\partial C}{\partial t} = a^2 \nabla^2 C - bC + R$

can be rewritten as $\dot{C} = M * C + R$

where

$$M = \frac{1}{h^2}\begin{bmatrix} 0 & a^2 & 0 \\ a^2 & -4a^2 - h^2 b & a^2 \\ 0 & a^2 & 0 \end{bmatrix}$$

Euler's Method

- We are really after C, rather than its derivative. Therefore, we must integrate the equation:

$$C_t = C_0 + \int_0^t \dot{C}\, dt$$

- The integration is performed using small discrete time steps:

$$C_{t+\Delta t} = C_t + \Delta t \left(M * C_t + R_t \right)$$

Anisotropic Diffusion

- More interesting pattern can be created if the diffusion rates are different in different directions. Instead of modeling diffusion with the Laplacian operator

$$a^2 \nabla^2 C = a^2 \left(\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2} \right)$$

- use different rates along X and Y directions:

$$a_1^2 \frac{\partial^2 C}{\partial x^2} + a_2^2 \frac{\partial^2 C}{\partial y^2}$$

More General Anisotropy

- Consider the Hessian matrix: $H = \begin{bmatrix} \frac{\partial^2 C}{\partial x^2} & \frac{\partial^2 C}{\partial x \partial y} \\ \frac{\partial^2 C}{\partial y \partial x} & \frac{\partial^2 C}{\partial y^2} \end{bmatrix}$
- X-Y anisotropy: $\dot{C} = \mathrm{Tr}\left(D^{\mathrm{T}} H D \right), \quad D = \begin{bmatrix} a_1 & 0 \\ 0 & a_2 \end{bmatrix}$
- General anisotropy: $\dot{C} = \mathrm{Tr}\left(D^{\mathrm{T}} Q^{\mathrm{T}} H Q D \right)$

- Euler integration is as before:

$$C_{t+\Delta t} = C_t + \Delta t\left(M * C_t + R_t\right)$$

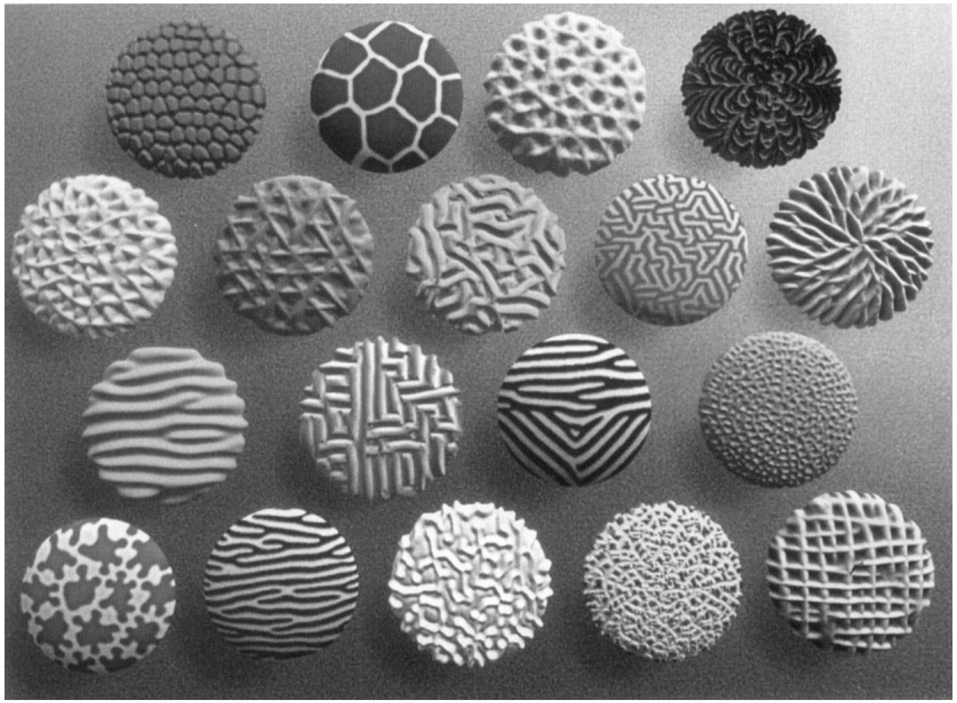
- Only the convolution mask has changed:

$$M = \frac{1}{2h^2}\begin{bmatrix} -a_{12} & 2a_{22} & a_{12} \\ 2a_{11} & -4(a_{11}+a_{22})-2h^2b & a_{11} \\ a_{12} & 2a_{22} & -a_{12} \end{bmatrix}$$

- where $A = Q^{\mathrm{T}} D^{\mathrm{T}} D Q$

Space-Varying Diffusion

- Control diffusion pattern spatially by specifying a "diffusion map" - a map that gives the diffusion coefficients at each position on the surface.

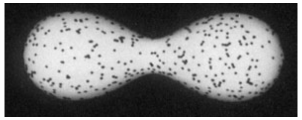


Reaction-Diffusion on Surfaces

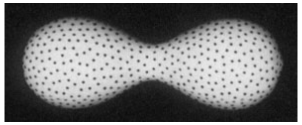
- To avoid the difficulties associated with mapping 2D textures onto objects in 3D, we'd like to perform the RD simulation directly on the surface of an object.
  - Randomly distribute points over the object's surface
  - Create a mesh of even-sized cells
  - Simulate R-D on this mesh

Relaxation of Random Points

- Given a polyhedral model, randomly place points (for uniform distribution, the probability of a point being placed in a polygon is proportional to its area).



- Relaxation: move each point according to repulsion forces applied to it by other nearby points (iterate several times).

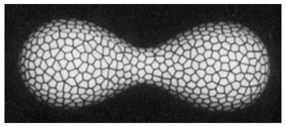


Relaxation

- Loop k times:
  - For each point P on surface
    - Determine nearby points to P
    - Map these points onto plane of P's polygon
    - Compute and store repulsive forces on P
  - For each point P on surface
    - Compute the new position of P based on repulsive forces

Mesh Construction

- Compute Voronoi regions, based on the points computed in the previous stage:



- The Voronoi region of a point P contains all points that are closest to P
- For each point we compute it's Voronoi neighbors and the lengths of the edges between adjacent Voronoi regions.

Reaction-Diffusion on a Mesh

- The edge lengths become the diffusion coefficients (normalized to sum to 1)
  - Results in isotropic diffusion.
  - How can anisotropic diffusion be accomodated?

- The Laplacian of a cell is computed taking into account all of the cells neighbors. Each neighbor's contribution is weighted by the diffusion coefficient corresponding to the edge between the two cells.

Rendering

- For rendering, the results of the R-D simulation are interpolated: