Reaction-Diffusion Textures

Andrew Witkin* and Michael Kass†

Keywords—Texture Synthesis; Natural Phenomena; Simulation.

Abstract

We present a method for texture synthesis based on the simulation of a process of local nonlinear interaction, called reaction-diffusion, which has been proposed as a model of biological pattern formation. We extend traditional reaction-diffusion systems by allowing anisotropic and spatially non-uniform diffusion, as well as multiple competing directions of diffusion. We adapt reaction-diffusion systems to the needs of computer graphics by presenting a method to synthesize patterns which compensate for the effects of non-uniform surface parameterization. Finally, we develop efficient algorithms for simulating reaction-diffusion systems and display a collection of resulting textures using standard texture- and displacement-mapping techniques.

1 Introduction

Texture mapping techniques have become so highly developed and so widely used that textureless images tend to appear barren, unrealistic, and boring. To date, though, techniques for *synthesizing* natural textures have advanced far less than texture rendering methods. A few noise-based textures, such as marble and fractal bumps, have become standard (see, e.g.,[12, 18, 17]), and specialized methods for synthesizing stone walls are presented in [13]. In [7], a statistical method is presented for encoding and reproducing natural textures, while [3] describes the use of fractal methods for statistical texture encoding. Even so, scanned real-world images still provide a principal source of realistic texture maps.

In this paper, we investigate a class of patterns that arise from local, nonlinear interactions of excitation and inhibition. Our starting point is a chemical mechanism that was first proposed by Alan Turing [23] to account for pattern formation in biological morphogenesis. The basis for Turing's idea is the notion that cell properties, such as pigment production, are fixed during the development of the embryo in a way that depends on the concentrations of one or more chemical messengers which he dubbed *morphogens*. He postulated that patterning is governed primarily by two concurrently operating processes: *diffusion* of morphogens through the tissue and chemical *reactions* that produce and destroy morphogens at a rate that depends, among other things, on their concentrations. Such mechanisms are called *reaction-diffusion* (RD) systems.

Reaction-diffusion systems give rise to nonlinear partial differential equations, in which the time derivative of morphogen concentration at each point in the medium is given as a function of the current concentration and of derivatives of concentration with respect to position. The nonlinear model largely defies analysis; its behavior must be understood through numerical simulation. In [23], Turing extensively analyzed a linear approximation to the nonlinear equation, but lacked the computing tools to attack the nonlinear model numerically.

Since Turing's initial proposal, mechanisms of this kind have been invoked to account for several biological patterns, including spotted and striped coats of cats, zebras and giraffes [1, 15, 2, 25, 24], the markings on certain butterfly wings [14], and the arrangement of ocular dominance columns in mammalian cerebral cortex [22]. Reaction-diffusion equations have also arisen in such diverse fields as image processing [19], population dynamics[6] and epidemiology[11]. In [9] we considered

*School Of Computer Science, Carnegie Mellon University, Pittsburgh PA 15213. aw@cs.cmu.edu

† Advanced Technnology Group, Apple Computer, Cupertino, CA 95014. kass@apple.com

Figure 1: Glass. Reaction-diffusion patterns are used to control opacity, displacement, and shading parameters of some glassware.

reaction-diffusion systems as a model for fingerprints.

Here we will address three main questions: (1) How can we extend the range of patterns we can create with reaction-diffusion systems? (2) How can we adapt reaction-diffusion systems to the needs of computer graphics? (3) How can we simulate the reaction-diffusion process efficiently? Our answers to these questions lead to the following specific contributions:

We generalize the basic RD model by introducing anisotropic and space-varying diffusion. The addition of anisotropy allows the creation of zebra stripes, and also, surprisingly, of sand ripples. Allowing diffusion rates and directions to vary over space creates more complex patterns, including the swirling patterns typical of fingerprints.

We consider new reaction function models, including functions that allow multiple competing orientations at each point. The addition of multiple competing directions gives rise to a host of new patterns, including some exhibiting a striking woven or lattice-like appearance.

We investigate new patterns that can be created by using non-standard initial conditions or reaction-diffusion parameters that change during the simulation. These include, for example, convincing giraffe markings.

We show how RD textures can avoid some standard difficulties of parametric texture mapping. First, by adjusting diffusion rates and directions, it is possible to grow textures that incorporate correction for parametric distortion, appearing uniform when applied to the surface. Second, texture patches can be seamlessly joined through the use of shared boundary conditions. Similarly, cyclic boundary conditions may be used to join a patch to itself, or to create seamless periodic textures.

We derive two fast but simple algorithms for the solution of the partial differential equations governing reaction-diffusion systems. The first makes use of the fact that the Green's function of the diffusion equation is a Gaussian and computes the effects of diffusion with efficient Gaussian convolution techniques. The second is a multigrid technique which simulates the equations rapidly on a coarse grid and then refines its results on a series of finer and finer grids.

2 Basic Reaction-Diffusion Dynamics

Our RD model incorporates three processes—diffusion, dissipation, and reaction. Diffusion governs the transport of morphogens from points of higher concentration to points of lower concentration. Dissipation involves the breakdown of morphogens, causing concentrations, in the absence of other influences, to decay exponentially toward zero. Reaction governs the rate of morphogen production. The effects of these three processes sum to provide the time derivative of concentration.

Since we are interested in generating surface textures, we assume throughout that diffusion occurs through a two-dimensional medium, although analogous results can be derived for arbitrary dimensions. The concentration of each of the morphogens is a function of position and of time. We denote the concentration function for a particular morphogen by $C(x, y)$. The reaction-diffusion equation governing the morphogen is then given by

$$\dot{C} = a^2 \nabla^2 C - bC + R, \tag{1}$$

where $\dot{C}$ is the time derivative of C, $\nabla^2 C$ is the *Laplacian* of C, defined by

$$\nabla^2 C = \frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2}, \tag{2}$$

a is the rate constant for diffusion, and b is the rate constant for dissipation. The function R is the reaction function governing C, which can depend on all the other concentrations. The three terms on the right hand side of equation 1 represent diffusion, dissipation, and reaction, respectively.

In practice, we represent the concentration functions as two-dimensional arrays of discrete samples. To evaluate $\dot{C}$ we approximate the space derivative $\nabla^2 C$ by a *finite difference*. The second finite difference in the x direction is

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

where the i's and j's are array subscripts, and h is the distance between adjacent samples. Taking the corresponding second difference in the y direction, and summing, gives

$$\nabla^2 C_{i,j} \approx \frac{C_{i+1,j} + C_{i-1,j} + C_{i,j+1} + C_{i,j-1} - 4C_{i,j}}{h^2}. \tag{3}$$

The discrete Laplacian can also be expressed as the convolution of the concentration array with the 3×3 mask

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

The values in the mask simply represent the coefficients in equation 3. The convolution form offers the advantage that any additional terms that are linear functions of $C_{i,j}$ and its neighbors may be readily combined to form a single 3×3 mask. Multiplying the Laplacian by a^2, and including the term $-bC_{i,j}$ gives the mask

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

in terms of which equation 1 becomes

$$\dot{C} = M * C + R,$$

using "$*$" to denote discrete convolution.

To compute C using 1, we must integrate $\dot{C}$ through time. The simplest integration formula, known as *Euler's method*, is

$$C_{t+\Delta t} = \Delta t(M * C_t + R_t), \tag{5}$$

which takes a timestep of size Δt.

3 Anisotropic Diffusion

The simple model of equation 1 assumes that diffusion occurs at a uniform rate in all directions and at all positions. In relaxing this restriction, we make it possible to produce a far wider range of patterns, including oriented patterns typical of zebra stripes or sand dunes.

Recall that the isotropic diffusion term of equation 1 is $\dot{C}_d = a^2(\partial^2 C/\partial x^2 + \partial^2 C/\partial y^2)$. To make C diffuse at different rates in x and y, we replace a^2 by independent rate constants for x and y, so that $\dot{C}_d = a_1^2 \partial^2 C/\partial x^2 + a_2^2 \partial^2 C/\partial y^2$. By varying a_1 and a_2, the RD pattern can be stretched or compressed, but only along the two coordinate axes.

To handle the general case, we introduce the *Hessian* matrix, defined by

$$H_{ij} = \frac{\partial^2 C}{\partial r_i \partial r_j},$$

where the vector $\mathbf{r} = [x, y]$. In terms of the Hessian, the isotropic diffusion term is $a^2 \operatorname{Tr}(H)$, where $\operatorname{Tr}(H)$, the *trace* of the Hessian, is the the sum of its diagonal elements. For the special case of anisotropic diffusion with axis-aligned principal directions, we can define the matrix

$$D = \begin{bmatrix} a_1 & 0 \\ 0 & a_2 \end{bmatrix},$$

in terms of which $\dot{C}_d = \operatorname{Tr}(D^T H D)$. We model diffusion with arbitrary principal directions by rotating the matrix D to bring the x and y axes onto the desired principal directions, giving the diffusion term $\dot{C}_d = \operatorname{Tr}(D^T Q^T H Q D)$, where Q is the rotation matrix.

Rather than working with this compound matrix directly, it is convenient to define the single *diffusion matrix*

$$A = Q^T D^T D Q,$$

in terms of which the diffusion is $\dot{C}_d = \sum_i \sum_j A_{ij} H_{ij}$. The diffusion matrix is given by

$$A = \begin{bmatrix} a_1^2 \cos^2\theta + a_2^2 \sin^2\theta & (a_2^2 - a_1^2)\cos\theta \sin\theta \\ (a_2^2 - a_1^2)\cos\theta \sin\theta & a_2^2 \cos^2\theta + a_1^2 \sin^2\theta \end{bmatrix} \tag{6}$$

where a_1 is the diffusion rate in the principal direction $[\cos\theta, \sin\theta]$ and a_2 is the diffusion rate in the principal direction $[-\sin\theta, \cos\theta]$.

When the Hessian is expressed in terms of finite differences, the quantity $\sum_i \sum_j A_{ij} H_{ij} - bC$, representing diffusion and dissipation, can be expressed as the convolution of C with a 3×3 mask which is a generalization of the isotropic mask given in equation 4. That mask is

$$M = \frac{1}{2h^2}\begin{bmatrix} -a_{12} & 2a_{22} & a_{12} \\ 2a_{11} & -4(a_{11} + a_{22}) - 2h^2 b & 2a_{11} \\ a_{12} & 2a_{22} & -a_{12} \end{bmatrix}. \tag{7}$$

where a_{11}, a_{12} and a_{22} are the three distinct elements of the symmetric matrix A.

The Euler update formula for anisotropic diffusion still has the form

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

but now the mask M is the one given in equation 7.

3.1 Space-varying diffusion

A diffusion matrix that is constant over position can only produce patterns whose direction and degree of elongation are constant as well. A further important generalization of the model is obtained by allowing the matrix A to vary with position. This is done by means of a *diffusion map*, an array that specifies the three distinct elements of A at each position. In practice, the diffusion map can be much coarser than the concentration map, with bilinear interpolation sufficing to obtain intermediate samples. Usually, the diffusion map is most naturally specified indirectly, by

giving $[\theta, a_1, a_2]$, or often just θ, as a function of position. Direction fields may in turn be created in a variety of ways—for example by interactive specification [20], by analyzing natural images [9], or through analytic forms.

4 Mapping onto surfaces

A well-known difficulty with parametric texture mapping is that textures undergo distortion in the mapping from parameter space to the surface. Although solid texture methods [18, 17] are not subject to this problem, they are not well suited to modeling textures that actually grow on surfaces, rather than in space. Generally, it is not possible to correct parametric distortion by inverse warping a texture after the fact. However, in this section, we show how RD textures may be *grown* in a way that incorporates the inverse warp by transforming the diffusion matrix. Like trick pictures that are meant to be viewed in curved mirrors, the resulting patterns appear grossly distorted when viewed in parameter space, but map correctly onto the surface.

The correction we will describe is based on a simplifying assumption that the parametric surface function is locally linear. Under this approximation, we correct fully for parametric stretch and shear, but not for distortion due to the second derivative of the parametric function. In our experience to date, this approximation has not produced visible artifacts. A related approach to the creation of inverse-warped statistical textures is describe in [7]. In a different approach, Turk [24] computes reaction diffusion textures directly on the vertices and edges of a polygonal mesh.

Previously, we described anisotropic diffusion in terms of $[\theta, a_1, a_2]$ where θ is the angle between the first principal direction and one of the texture coordinate axes, and a_1 and a_2 are the rates in the principal directions. We wish to use essentially the same description for diffusion on a surface, except that θ is to be interpreted as the angle on the tangent plane of the surface between one of the principal diffusion directions and an arbitrary reference direction, which could conveniently be chosen to be one of the two parametric directions. Of course a_1 and a_2 should describe the desired principal diffusion rates on the surface, not in parameter space.

If the parametric surface function is $\mathbf{x}(\mathbf{u})$, then the *Jacobian* matrix, $J = \partial\mathbf{x}/\partial\mathbf{u}$, serves as a basis for the surface's tangent plane at $\mathbf{x}$, i.e. $\delta\mathbf{x} = J\delta\mathbf{u}$ is a tangent vector. First-order distortion arises because J generally is not orthonormal, so lengths and angles are not preserved. We remove the distortion by performing a change of variables from $\mathbf{u}$ to $\mathbf{v}(\mathbf{u})$ such that $\partial\mathbf{x}/\partial\mathbf{v}$ is an orthonormal basis. To apply the correction, we then post-multiply the diffusion matrix A by the inverse of the matrix $\partial\mathbf{v}/\partial\mathbf{u}$, and pre-multiply by the inverse transpose.

To orthonormalize J, we must find a 2×2 matrix V such that

$$V^T J^T J V = V^T M V = I,$$

where $M = J^T J$ is the *metric tensor* of the surface, and I is the identity matrix. Then JV is by definition orthonormal. Orthonormal bases are only unique down to a rotation, so we must also pick an arbitrary reference direction on the surface with respect to which θ will be measured. Letting $\alpha = [1, 0]$ and $\beta = [0, 1]$, we choose $J\alpha$ as the reference direction, giving the additional condition on V that $\beta^T V \alpha = 0$, which simply means $v_{21} = 0$. This leaves a quadratic system to solve for the remaining components of V. Solving and inverting the matrix gives, in terms of the components of M,

$$\tilde{V} = \frac{1}{\sqrt{m_{11}}} \begin{bmatrix} m_{11} & m_{12} \\ 0 & \sqrt{m_{11}m_{22} - m_{12}^2} \end{bmatrix}. \quad (8)$$

In summary, given the desired θ relative to the $[1, 0]$ parameter direction on the surface and the principal rates a_1 and a_2, the corrected diffusion matrix $\hat{A}$ can be computed in the following steps:

- Compute the uncorrected diffusion matrix A, in terms of $[\theta, a_1, a_2]$, according to equation 6.
- Compute the surface Jacobian $J = \partial x/\partial u$.
- Compute the metric tensor $M = J^T J$.
- Evaluate $\tilde{V}$ according to equation 8.
- Obtain the corrected diffusion matrix by evaluating $\hat{A} = \tilde{V}^T A \tilde{V}$.

The 3×3 convolution mask is then computed, and used according to equation 5 to take time steps. Figure 2 shows comparable textures, in parameter space and on the surface, with and without the correction described in this section.

4.1 Sewing patches together

It is often difficult or impossible to describe a complex surface using a single parametric function. Even simple surfaces, such as the sphere, cannot be parameterized without introducing singularities. These problems cannot be solved using the correction described above. However, it should be possible to solve them using piecewise parameterizations, letting each patch provide boundary conditions for its neighbors. Differential geometry provides a formalism for piecewise parameterizations in the construct of *coordinate charts and atlases* [21]. Roughly speaking, a chart is a parameterization for a piece of the surface, and

Figure 2: Correcting parametric distortion. The object on the left has been mapped with an uncorrected texture, showing marked parametric distortion. The object on the right has been mapped with a corrected texture, synthesized to appear undistorted on the surface. The raw textures appear behind each object.

an atlas contains sufficient overlapping charts to cover the surface, together with functions that map among the charts where they overlap. For example, the faces of a cube can serve as an atlas for a sphere. Although we have not yet implemented this scheme for curved surfaces, we have tested the use of boundary conditions to join planar patches and to create seamless periodic texture using cyclic boundary conditions. Figure 3 includes such a periodic texture.

5 Efficient Solution Methods

The Euler simulation method of section 2 is adequate for low-resolution simulations, but its computational complexity is unfortunately at least $O(r^4)$ where $r = 1/h$ is the sampling rate or resolution of the pattern. In order to improve on this bound, we develop two alternative algorithms capable of achieving $O(r^2)$ performance. The first of these algorithms is based on Gaussian convolutions, while the second is a *multi-grid* algorithm.

While the full non-linear reaction-diffusion equation is very difficult to analyze, we can gain substantial insight into the complexity of the computation by considering the linear differential equation which arises when $R = 0$ in equation 1 and the matrix A is constant. For this special case of pure diffusion and dissipation, there exists an analytic form for the solution [10]:

$$C_{t+\Delta t}(x, y) = C_{t_0}(x, y) * G_{\Delta t}$$

where

$$G_{\Delta t}(x, y) = -\frac{1}{4\pi\Delta t\sqrt{\det(A)}} \exp\left(\frac{-p^T A p}{4\Delta t \det(A)} - bt\right) \tag{9}$$

is known as the Green's function of the differential equation. The solution is the convolution of the initial conditions with a Gaussian that has been aligned with the principal axes of the matrix A and scaled to take account of the dissipation rate. The size of the Gaussian is proportional to $\sqrt{\Delta t}$.

Consider trying to compute the above solution with the Euler method of section 2. Each iteration is a convolution of the initial conditions with a 3×3 kernel. Variances add under repeated convolution [4], so after n iterations, the resulting standard deviation of the Gaussian will be proportional to $\sqrt{n}$. Achieving a fixed kernel size therefore takes $O(r^2)$ iterations. Since each iteration takes $O(r^2)$ work, the total computation is $O(r^4)$. Clearly this can be a serious obstacle to synthesizing high-resolution patterns.

5.1 Gaussian Convolution

Instead of simulating the large-kernel convolution with a series of small convolutions, we can simply convolve G with C and get the same result. Using a direct spatial convolution also results in $O(r^4)$ complexity, but there are several ways of doing the convolution more efficiently. Two-dimensional Gaussians are separable, so they can be factored into the product of one-dimensional Gaussians. This makes it possible to convert the two-dimensional convolution into two one-dimensional convolutions and thereby reduce the complexity to $O(r^3)$. If each of these one-dimensional convolutions is approximated with recursive filters[16], the complexity is further reduced to $O(r^2)$. Another alternative is to use hierarchic convolutions [5] which achieve $O(r^2)$ complexity by using image pyramids.

While the above techniques efficiently compute the effects of diffusion and dissipation alone, we still need to incorporate the effects of reaction in order to simulate the full differential equation. Using the Green's function again, we can write the solution to the full differential equation as follows:

$$C_{t_0+\Delta t} = C_{t_0} * G_{\Delta t} + \int_0^{\Delta t} (R * G_u)\, du. \tag{10}$$

We already know an efficient way to compute the first term on the right in equation 10. Now we seek a simple, efficient approximation to the second term in order to complete the solution method. One way to do this is to use a constant approximation $\hat{R}(x, y, t) = R(x, y, t_0)$ leading

to the iterative formula

$$C_{t_0+\Delta t} = C_{t_0} * G_{\Delta t} + \hat{R} * \int_{t_0}^{t_0+\Delta t} G_u \, du, \qquad (11)$$

where it remains to evaluate the definite time integral of the Green's function.

Unfortunately, G is not integrable, so even the piecewise constant approximation to R is somewhat problematic. We have three natural choices. First, we can perform the integral numerically. Second, it *is* possible to obtain $\int_0^\infty G \, dt$ by means of *Hankel Functions* [10], allowing us to solve each constant approximation to equilibrium. Third, and by far the simplest, we can render the integral trivial by approximating R as an *impulse* applied at $t = t_i$. This option leads directly to the algorithm

$$C_{t_0+\Delta t} = \left[C_{t_0} + \gamma R_{t_0}\right] * G_{\Delta t}, \qquad (12)$$

where γ is a factor intended to correct for R's application at the beginning of the step, rather than throughout. To make the space integral of C after the step be the same as it would have been under a constant, rather than impulsive, approximation to R, this factor should be

$$\gamma = \frac{e^{b\,\Delta t} - 1}{b}.$$

The algorithm that results from this approximation allows for diffusion with $O(r^2)$ computation using hierarchic convolutions or separable recursive approximations to Gaussians. The chief limitation is that the technique only works when the matrix A does not vary over space. Subject to this limitation, however, it offers a potentially dramatic performance advantage over the methods that have been used previously, particularly for large step sizes or high diffusion rates. Swindale [22] arrived at a similar algorithm by a different path in the course of modeling the formation of ocular dominance columns.

5.2 Multi Grid

Another approach to avoiding the $O(r^4)$ behavior of the Euler simulation is to use multi-grid techniques[8]. The basic idea of these techniques is to simulate the equations on a series of grids of different sizes. Iterations on the coarser grids allow the simulation to proceed very rapidly, while iterations on the finer grids provide the detail necessary in the final result. If the grid sizes differ by a factor of two in each dimension from one level to the next, then the total number of samples in the grids is approximately $4/3$ as many as in the finest grid. As a result, a constant number of iterations on each of the grids can be accomplished in $O(r^2)$ time.

In some contexts, effective multi-grid implementations require complex control strategies to switch from one grid level to another. In order to synthesize reaction-diffusion textures, however, we have found a simple strategy to be very effective. We begin with the coarsest grid able to resolve details at the intrinsic scale of the pattern being synthesized. We do a small fixed number of iterations on that grid and then interpolate the result onto the next finer grid. This is repeated for all the remaining grids. Iterations on the different grids make use of the same 3×3 matrix M as the Euler method. The only change is that the grid-spacing h changes as we move from one grid to another. The resulting multi-grid method can be as efficient as the Gaussian convolution method, but is not subject to the limitation that the matrix A be constant.

5.3 Binary Convolution

Convolutions of binary data with an integer-valued mask can be performed efficiently by means of lookup tables: accessing eight consecutive binary data points as a byte, the sum of the eight corresponding convolution values can be obtained in a single table lookup. Although $O(r^4)$, binary convolution may nevertheless be the fastest option for convolution with small masks. For reaction functions that are expressible in binary form, the algorithm of equation 12 may be reduced to one of iterated binary convolution by allowing the diffusion constant, b to grow very large, rendering the contribution of the old concentrations negligible. In the resulting simplified algorithm, the binary reaction function is applied to the concentrations, with binary convolution applied to the result. Binary convolution has been used previously to compute RD patterns by Young [25].

6 Results

Although we have described one basic pattern-forming mechanism, a wide variety of RD textures can be produced by varying the initial and boundary conditions, the number of morphogens involved, the rate constants for each, the reaction functions that govern their interactions and the manner in which concentrations are mapped into surface appearance. In this section we show the textures we have produced and describe the choices that led to their creation.

All of the images that accompany this paper were rendered using Photorealistic RenderMan. RD textures were used as displacement and texture maps. In most images, monochrome concentration maps were used to blend between pairs of colors or other parameters such as opacity and specularity.

6.1 Isotropic Patterns

To create a simple isotropic RD pattern, we can use two concentration arrays C^+ and C^-, with different diffusion

rates a^+ and a^-, producing convolution masks M^+ and M^- according to equation 4. A threshold on the difference $C^+ - C^-$ serves as the reaction function:

$$R^+ = R^- = \text{if } (C^+ > C^-) \text{ then } k \text{ otherwise } 0 \quad (13)$$

where k is a reaction constant. Starting this RD system from initial conditions which are zero except on a jittered diamond grid produces a roughly cellular pattern that strikingly resembles giraffe markings (figure 3-1b). A similar pattern (3-1a), finer in scale and rendered with displacement mapping, looks reptilian. Varying the reaction rate constant k (from equation 13) over time as the pattern is formed produces very different results (3-4a, 3-4c).

6.2 Simple Anisotropy: Stripes

The simple two-morphogen system used to produce the giraffe markings involves two morphogens diffusing isotropically, but at different rates. Introducing anisotropy allows more freedom. If we describe anisotropic diffusion using triples $P = [\theta, a_1, a_2]$, for each concentration map C, then branching and merging stripe patterns are produced with $P^+ = [\theta, es, s]$, and $P^- = [\theta, es, ds]$, which makes both morphogens diffuse at the same rate in the θ direction, but different rates in the perpendicular direction. Satisfactory zebra-like stripes can be produced with $e = 2.0$, $d = 1.5$, $s = 1.0$, $b = 1.0$, $k = 1.0$, $h = 1.0$, $\Delta t = .02$, and C^+ and C^- initialized to random noise in the interval $[-1, 1]$. A typical zebra pattern is shown in figure 3-4b. A very similar RD pattern, rendered as a displacement map with appropriate coloring, resembles rippled sand. The addition of a fine noise-based grain texture (figure 5) enhances the effect.

6.3 Competing Orientations: weaves, lattices, and mazes

The interesting behavior of the anisotropic system described above led us to consider more elaborate systems involving multiple orientations. The general idea we considered was to replace the single antagonistic pair of morphogens with a set of such pairs, each pair differently oriented. Thus, a two-direction system might involve an antagonistic pair of morphogens biased toward horizontal diffusion, as well as a vertically biased pair.

Within this framework we discovered two reaction functions that produce particularly interesting patterns. (We denote by D_i the difference $C^+ - C^-$ for the ith pair of concentrations.) The first is a "max of differences" (M-D),

$$\begin{aligned} R &= k, \quad \text{if} \quad \max_j D_j > \rho, \\ &= 0, \quad \text{otherwise.} \end{aligned} \quad (14)$$

where ρ is a threshold, and the second, a "difference of abs's" (D-A):

$$\begin{aligned} R &= k, \quad \text{if} \quad \text{abs} \max_j D_j > \text{abs} \min_j D_j \\ &= 0, \quad \text{otherwise.} \end{aligned} \quad (15)$$

Although the D-A and M-D reaction functions appear similar, the patterns they produce could hardly be more different. Figure 3-3b shows a two-direction D-A texture, which has a maze-like appearance. In contrast, a two-direction M-D texture (3-4e) appears woven, although such irregularities as splitting and merging of "threads" give it a distinctly organic appearance. Figures 3-2b and 3-1c show three- and five- direction M-D textures, while figure 3-2c shows a five-direction D-A texture. All of these derive from uniform random initial conditions.

6.4 Spatial Variation

Figures 3b, 4b, 4c, and 4d show RD patterns governed by diffusion maps. Figure 3-3d is a two-orientation M-D pattern grown on a radial/concentric orientation map. Figure 3-2a shows a similar pattern, but with rotation of the map orientations to produce a double spiral. Figure 3-2e shows two competing nearly-radial orientations. Figure 3-3c shows the merger, with smoothing, of three uniform orientation fields, resembling a zebra's haunch markings. Figures 3-1d and 3-4d show more complicated orientation patterns. All of these diffusion maps were generated using simple analytic forms.

6.5 Boundary Conditions

An unusual feature of RD textures is that texture patches that join seamlessly can be created by copying data across the boundaries at each iteration. This is illustrated in figure 3-2d: a texture was grown using doubly cyclic boundary conditions, giving it the topology of a torus. This was accomplished simply by wrapping the array references across both horizontal and vertical boundaries while calculating the convolution $C * M$. In the resulting pattern, the left edge joins smoothly to the right, and the upper edge to the lower. The figure shows how the replicated pattern tiles the image without seams. The replication itself is easily perceived, due to the repetition of gross features within the texture. However, the seams between the tiles are invisible.

7 Conclusion

Texture synthesis is a hard problem because the range of natural textures, and of texture-forming processes, is vast. Given this diversity, we cannot expect to find a single, universal texture generator. Nonetheless, by identifying

processes that are widespread in nature, we can develop models that are broadly useful for texture synthesis.

Although RD models have been applied to problems in a variety of fields, their use in computer graphics is in its infancy. Already, RD models significantly extend the range of textures that can be synthesized. Their characteristic organic appearance lends an interesting new element to synthetic imagery. In addition, RD textures have the potential to extend the range of surfaces to which textures can be applied because they can be grown to compensate for parametric distortion and joined smoothly patch to patch.

Acknowledgements

The research reported in this paper was conducted at Carnegie Mellon University, and at Apple Computer. At Carnegie Mellon, work was supported in part by grants from Apple Computer and from the Siemens Corporation, and by an equipment grant from Silicon Graphics, Inc. Photorealistic RenderMan software, which was used to render all the images appearing in this paper, was provided to Carnegie Mellon through Pixar's RenderMan Education Program. The "Space Cookies" and "Fungi" images were created by Drew Olbrich of Carnegie Mellon. Drew Olbrich, Michael Gleicher, William Welch, and Wendy Plesniak provided valuable assistance.

References

[1] Jonathan Bard and Ian Lauder. How well does turing's theory of morphogenesis work? *Journal of Theoretical Biology*, 45:501–531, 1974.

[2] Jonathan B.L. Bard. A model for generating aspects of zebra and other mammalian coat patterns. *Journal of Theoretical Biology*, 93:501–531, 1981.

[3] Michael Barnsley. *Fractals Everywhere*. Academic Press, San Diego, 1988.

[4] R. N. Bracewell. *The Fourier Transform and its Applications*. McGraw-Hill, New York, 1986.

[5] P. J. Burt. Fast hierarchical correlations with gaussian-like kernels. Technical Report TR 860, Dept. of Computer Science, U. of Maryland, 1980.

[6] S. Ei and M. Mimura. Pattern formation in heterogeneous reaction-diffusion-advection systems with an application to population dynamics. *SIAM J. on Mathematical Analysis*, 21(2):346–361, 1990.

[7] Andre Gagalowicz and Song De Ma. Sequential synthesis of natural textures. *CVGIP*, 30:289–315, 1985.

[8] W. Hackbusch. *Multi-Grid Methods and Applications*. Springer-Verlag, New York, 1985.

[9] Michael Kass and Andrew Witkin. Analyzing oriented patterns. *Computer Vision, Graphics and Image Processing*, 37:362–385, 1987.

[10] Granino Korn and Thresa Korn. *Mathematical Handbook for Scientists and Engineers*. McGraw Hill, New York, 1968.

[11] K. Kunish and H. Schelch. Parameter estimation in a special reaction-diffusion system modelling man-environment diseases. *Journal of Mathematical Biology*, 27(6):633–665, 1989.

[12] B. Mandelbrot. *Fractals: Form, Chance, and Dimension*. W.H. Freeman, San Fransico, 1977.

[13] Kazunori Miyata. A method of generating stone wall patterns. *Computer Graphics*, 24(4):387–394, 1990.

[14] J. D. Murray. On pattern formation mechanisms for lepidopteran wing patterns and mammalian coat markings. *Philosophical Transactions of the Royal Society (B)*, 295:473–496, 1981.

[15] J. D. Murray. A pre-pattern formation mechanism for animal coat markings. *Journal of Theoretical Biology*, 88:161–199, 1981.

[16] A. Openheim and R. Schafer. *Digital Signal Processing*. Prentice-Hall, Englewood Cliffs, New Jersey, 1975.

[17] Darwyn R. Peachey. Solid texturing of complex surfaces. *Computer Graphics*, 19:279–286, 1985.

[18] Ken Perlin. An image synthesizer. *Computer Graphics*, 19:287–296, 1985.

[19] C. Price, P. Wambacq, and A. Oosterlinck. Applications of reaction-diffusion equations to image processing. In *Third Int. Conf. on Image Processing and its Applications*, pages 49–53, 1989.

[20] Karl Sims. Leonardo's deluge (video). *Siggraph '89 Computer Graphics Theater*, 1989.

[21] M. Spivak. *A Comprehensive Introduction to Differential Geometry (5 vols)*. Publish or Perish Press, 1975.

[22] N.V. Swindale. A model for the formation of ocular dominance stripes. *Philosophical Transactions of the Royal Society (B)*, 208:243–264, 1980.

[23] Alan Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society (B)*, 237:37–72, 1952.

[24] Greg Turk. Generating synthetic textures using reaction-diffusion. Technical Report TR-90-018, University of North Carolina, Chapel Hill, 1990.

[25] David Young. A local activator-inhibitor model of vertebrate skin patterns. *Mathematical Biosciences*, 72(1), 1984.

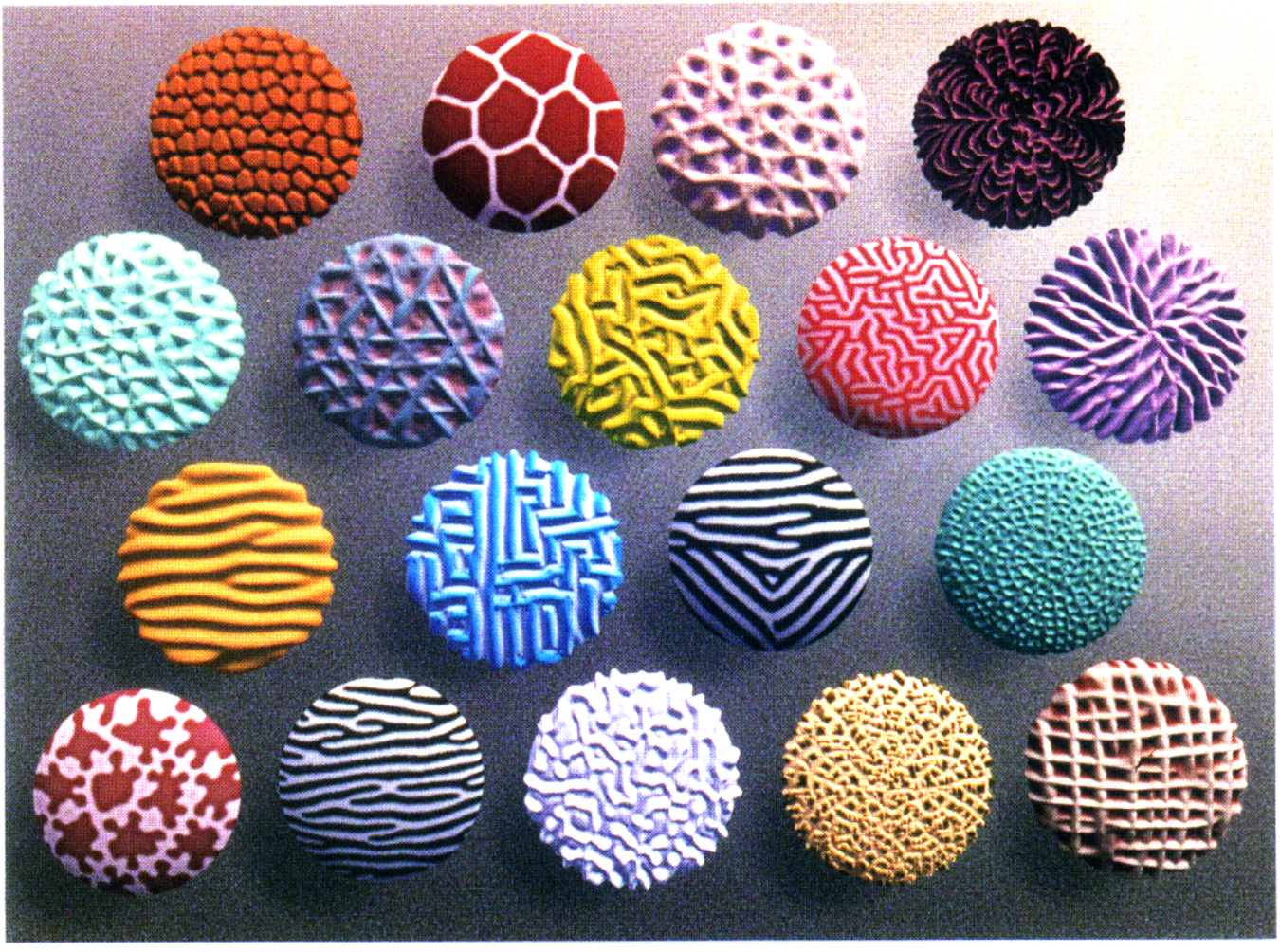

Figure 3: **Texture Buttons.** **Row 1:** (a) reptile, (b) giraffe, (c) coral, (d) scalloped. **Row 2:** (a) spiral, (b) triweave, (c) twisty maze, (d) replication, (e) purple thing. **Row 3:** (a) sand, (b) maze, (c) zebra haunch, (d) radial. **Row 4:** (a) space giraffe, (b) zebra, (c) stucco, (d) beats us, (e) weave.

By type. Isotropic: reptile, giraffe, space-giraffe, stucco. Multi-orientation: coral (5), spiral (2), triweave (3), twisty maze (5), replication (3), purple thing (2), maze (2), radial (2), beats us (2), weave (2). Diffusion mapped: scaloped, spiral, purple thing, zebra haunch, radial, beats us.

Figure 4: Space Cookies. All texture and displacement maps, except those on the bag and the plate, are RD patterns.

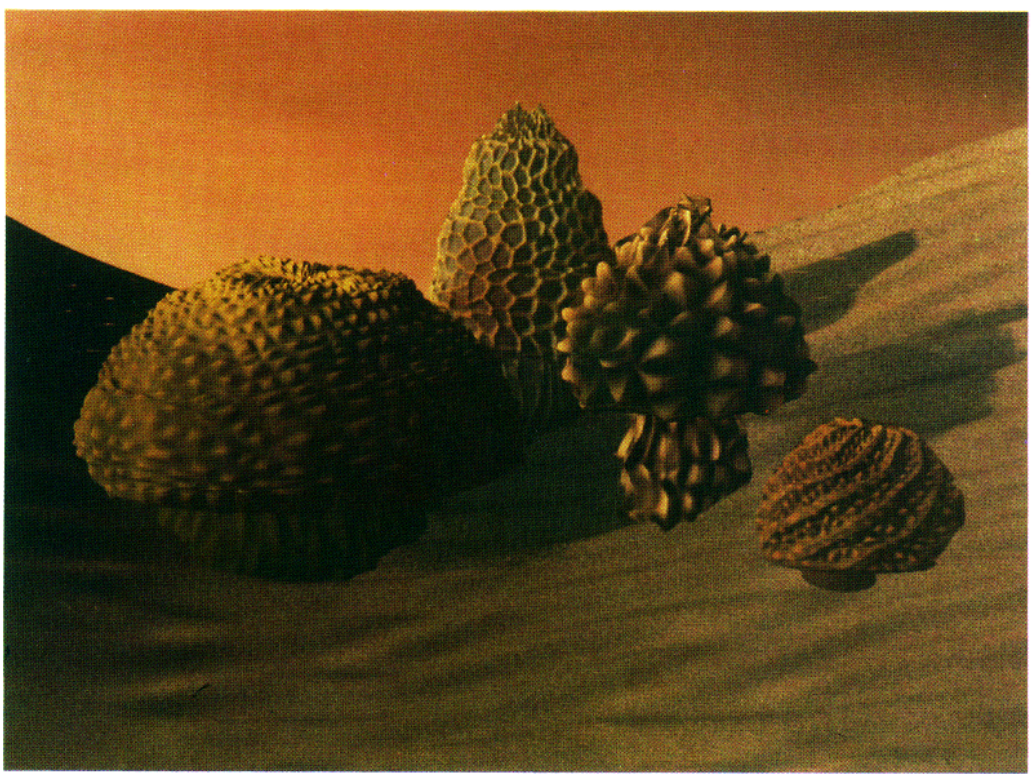

Figure 5: Fungi. All texture and displacement maps are RD patterns, except the sky, the fine grain pattern on the sand, and the fluting on the front right mushroom.