Generalized Reaction-Diffusion Textures

Tim McGraw

Computer Science and Electrical Engineering Department
West Virginia University

Abstract

We present a method of synthesizing textures based on a modified reaction-diffusion equation. A non-Gaussian model of diffusion is employed to make a new class of textures possible. Whereas the Gaussian diffusion model is characterized by its covariance matrix (a rank-2 tensor), the generalized model of diffusion is characterized by a sequence of tensors of increasing rank which represent higher-order moments of the diffusion displacement probability. A numerical method of solving the new reaction-diffusion equation is described, as well as representative textures generated by this technique. The resulting patterns are inorganic in nature, often featuring sharp corners. The generalized reaction-diffusion textures display spatial inhomogeneity, even when the diffusion process is homogeneous, making it possible to generate complex textures from few parameters when compared with previous techniques employing inhomogeneous reaction-diffusion. The preferred orientations depend on the tensor sequence, and can be inspected prior to texture generation by plotting the non-Gaussian diffusion propagator.

Key words: reaction-diffusion, texture synthesis, generalized diffusion, high-rank tensor

1 Introduction

Algorithms for automatic generation of models and textures is becoming more important as graphics applications are capable of displaying increasingly large and complex scenes. There have been many approaches [1] to synthesis of textures, including combinations of noise functions [2,3], basis functions based on feature point distances [4], Fourier synthesis [5], simulation of cellular development [6], exemplar-based texture synthesis [7,34–36] and reaction-diffusion textures.

Alan Turing [8] proposed the reaction-diffusion process as a model of pattern formation in animals. The process models the chemical reaction between two

Preprint submitted to Elsevier *31 August 2007*

"morphogens" and the simultaneous diffusion of those chemicals within a developing organism. Witkin and Kass [9] presented a technique for numerically simulating the reaction-diffusion equations in a Euclidean domain to generate textures. Greg Turk [10] described a simulation of the same process on manifolds, so that the texture evolves naturally over a surface. The original process described by Turing was homogeneous and isotropic. Witkin and Kass simulated inhomogeneous and anisotropic reaction-diffusion. Their governing parameters were allowed to vary spatially, and the diffusion process could be steered by use of a matrix within the diffusion term of the equation. This simulated a more general diffusion process, one that can vary spatially, but only in a limited way. At each location the diffusion process had a single preferred orientation. This orientation corresponds to the modes of the molecular displacement probability, in this case, a Gaussian distribution. In this paper we propose to generate textures by extending the diffusion process to be non-Gaussian. This process will be governed by a *sequence* of tensors of increasing rank. This allows the underlying diffusion displacement probability to be multimodal. This permits the generation of patterns with multiple preferred orientations, even when the diffusion process is homogeneous. This diffusion process corresponds to what would take place in a structured medium and makes a new class of textures possible. In subsequent sections we will present the theory, implementation and our results.

1.1 The Reaction-Diffusion Equation

The general reaction-diffusion equation for an interacting pair of chemicals can be written as the coupled system

$$\begin{aligned}\frac{\partial u}{\partial t} &= d_u \nabla^2 u + f(u,v), \\ \frac{\partial v}{\partial t} &= d_v \nabla^2 v + g(u,v).\end{aligned} \tag{1}$$

The quantities u and v represent the concentration of the two chemicals. It can be seen from (1) that the time rate of change of concentration depends on a diffusion process (the first term on the right-hand side) and reaction functions, designated $f(u,v)$ and $g(u,v)$. It is common to start the evolution of (1) from some initial condition which is near a steady-state solution and allow the simulation to run until convergence. The solution of the system will yield two scalar-valued images, u and v. Color textures are obtained by defining a palette to map concentration values to colors.

The choice of reaction functions has a large impact on the appearance of the resulting textures. In addition to Turing, other researchers have proposed reaction functions to model various classes of textures, such as those appearing on seashells [11], various mammals [12], reptiles [13] and fish [14]. The morphogens can be classified by inspecting the partial derivatives of f and g. Gierer and Meinhardt [15] described and categorized various reaction-diffusion systems based on the morphogen types. Models of reaction abound in the computational biology and chemistry literature, but to date the only modifications to the diffusion term have been to make it inhomogeneous and anisotropic.

Harris et al. [16] and Sanderson et al. [17] have implemented reaction-diffusion texture generation on the GPU to greatly increase the speed of the simulation. The time required to generate reaction-diffusion textures has been a barrier to their widespread use in computer graphics, though the model is common in physical chemistry and developmental biology. We will review several reaction functions, but our model is not dependent on any particular choice of reaction equation. Our work involves only the diffusion term in the reaction-diffusion equation and will work in conjunction with any choice of reaction function.

The reaction-diffusion model proposed by Turing,

$$\begin{aligned} \frac{\partial u}{\partial t} &= d_u \nabla^2 u + uv - u - \alpha \qquad (2) \\ \frac{\partial v}{\partial t} &= d_v \nabla^2 v + \beta - uv, \end{aligned}$$

is classified as an activator-substrate system. Depending on the reaction parameters α and β, it generates textures that resemble such patterns as leopard spots and zebra stripes. Note that for $u = 4$, $v = 4$, $\alpha = 12$, $\beta = 16$, the pattern is unchanging. The reaction rate is zero for this combination of values, and the diffusion rate is zero since the concentration gradient is zero everywhere. This is a steady-state solution to the system of equations, albeit an uninteresting one. Initializing the system to this state, then perturbing the system with random noise on the concentrations or parameters (or both), may result in other steady states which correspond to visually interesting patterns.

A different family of textures can be generated with the model presented by Gray and Scott [18]. The system of equations is given by

$$\frac{\partial u}{\partial t} = d_u \nabla^2 u - uv^2 + F(1 - u) \qquad (3)$$

$$\frac{\partial v}{\partial t} = d_v \nabla^2 v + uv^2 - (F + k)v, \qquad (4)$$

where F and k are user-chosen parameters. This system has a trivial steady state of $u = 1$, $v = 0$ for all parameter values. Initialization is performed by putting the system into a state nearby this trivial solution and perturbing the concentration values near the center of the array.

Pearson [19] mapped the parameter space of this system by generating images for many pairs of values of F and k, also identifying regions of stability and instability. Within the region of instability, no steady state is achieved so the texture continues to evolve. Sanderson et al. [17] discuss parameter setting for these and several other models.

1.2 The Diffusion Process

The simple model for diffusion used in Equation (1) is isotropic. This is a suitable model for free diffusion, such as may occur in the middle of a body of water. However, the model is not accurate when an impermeable membrane is present. Molecules will move freely *along* the membrane, but are restricted from diffusing *across* it. This can be modeled by linear transformation of the concentration gradient. The resulting anisotropic diffusion equation is given by

$$\frac{\partial u}{\partial t} = \mathrm{div}(D \nabla u). \tag{5}$$

The matrix, D, called the "diffusion tensor" is constrained to be symmetric and positive-definite. Symmetry maintains the diffusive property of the molecular transport. A general tensor can be decomposed into a sum of symmetric and antisymmetric tensors. The antisymmetric part is associated with advection [20] in Equation (5), not diffusion. Positive-definiteness of the diffusion tensor implies that diffusivity is positive (flux goes from high to low concentration regions). Anisotropic reaction-diffusion simulation will generate patterns with constant elongation and orientation [9], unless the diffusion tensor, D, varies spatially. We will demonstrate that this is not the case with higher rank tensors. Our textures will be characterized by one of several orientations at each point. The number of possible orientations increases with the number of terms in the expansion.

Isotropic, inhomogeneous diffusion is described by the equation

$$\frac{\partial u}{\partial t} = \mathrm{div}(d(x,y)\nabla u) = d(x,y)\nabla^2 u + \nabla d(x,y) \cdot \nabla u. \tag{6}$$

The additional term on the right-hand is due to the inhomogeneity of the diffusivity function $d(x,y)$. Further examples of inhomogeneous diffusion are discussed in section (3). Reaction-diffusion textures can be made inhomogeneous [9] by specifying a function $d(x,y)$ or a tensor field $D(x,y)$, and solving equation (6). Competing orientations can be achieved in this case by coupling multiple reaction-diffusion systems with differing rank-2 tensor fields. Since the model we present in this paper can generate inhomogeneous textures even when the diffusion process is homogeneous, we eliminate the process of designing the function $D(x, y)$ and can instead solve a simpler system of equations governing only two morphogens. Inhomogeneity in texture generation can also be introduced by allowing the reaction parameters to vary spatially. In this way, textures which transition from stripes to spots, for example, can be produced [17].

Inhomogeneous anisotropic diffusion has previously been used for vector field visualization [21,39] and tensor field visualization [22,38]. Vector field visualization involves constructing a field of tensors such that the principal eigenvector is parallel with the vector at each point. A single species reaction-diffusion process is simulated, starting from a random concentration function. The reaction function drives the concentration values toward one of two stable values, resulting in a high contrast image where the local texture orientation is parallel to the given vector field. Weickert [23] described the use of this equation to perform image denoising.

The diffusion transport process can be characterized by the diffusion propagator : the displacement probability of a molecule. If a molecule is located at position $\mathbf{x}_0$ at time t, then the probability that the molecule is at location $\mathbf{x}_0 + \mathbf{r}$ at time $t + \tau$ is denoted $p_\tau(\mathbf{r})$. In Gaussian diffusion the propagator is a Gaussian distribution with zero mean and covariance proportional to the diffusion tensor, $p_\tau(\mathbf{r}) = N(0, 2\tau D)$. The Gaussian diffusion propagator is unimodal and symmetric about the mean. In contrast, the diffusion propagator in our model can have multiple modes, and even nonzero skew and kurtosis, unlike the Gaussian distribution. Physically, generalized diffusion (or hyperdiffusion [24]) corresponds to restricted diffusion, such as may occur in crystalline or fiberous materials and other media with orientational heterogeneity.

2 Kramers-Moyal Expansion of the Diffusion Equation

In the previous sections we have seen the connection between the covariance matrix of the Gaussian diffusion propagator and the diffusion tensor D which appears in Equation (5). In this section we will see how higher order moments of a non-Gaussian diffusion propagator can be used to derive a more general diffusion equation.

Einstein's formulation [25] for the concentration, C, of particles undergoing

Brownian motion in 1-D,

$$C(x,t+\tau) = \int p_\tau(r)C(x-r,t)dr, \tag{7}$$

was based on Bachelier's earlier work on random walks. Expanding $C(x-r,t)$ in a Taylor series about $C(x,t)$ we obtain

$$C(x-r,t) = C(x,t) - r\frac{\partial C}{\partial x} + \frac{r^2}{2}\frac{\partial^2 C}{\partial x^2} - \frac{r^3}{3!}\frac{\partial^3 C}{\partial x^3}... \tag{8}$$

Substituting (8) into (7) we see that

$$\begin{aligned} C(x,t+\tau) = {} & C(x,t)[\int p_\tau(r)dr] - \frac{\partial C}{\partial x}[\int p_\tau(r)rdr] \\ & + \frac{1}{2}\frac{\partial^2 C}{\partial x^2}[\int p_\tau(r)r^2dr] - \frac{1}{3!}\frac{\partial^3 C}{\partial x^3}[\int p_\tau(r)r^3dr]... \end{aligned} \tag{9}$$

Note that the quantities in square braces are moments of the distribution $p_\tau(r)$. Since $\int p_\tau(r)dr = 1$, and using the fact that

$$\lim_{\tau\to 0}\frac{C(x,t+\tau)-C(x,t)}{\tau} = \frac{\partial C}{\partial t} \tag{10}$$

then for $\tau \ll 1$

$$\frac{\partial C}{\partial t} = -\frac{\partial C}{\partial x}\frac{<r>}{\tau} + \frac{\partial^2 C}{\partial x^2}\frac{<r^2>}{2\tau} - \frac{\partial^3 C}{\partial x^3}\frac{<r^3>}{3!\tau}... \tag{11}$$

where angle brackets ($< ... >$) denote moments. Truncating this series at the second term, allowing the first term (known as drift) to be zero, yields the familiar isotropic diffusion equation. A similar moment expansion of Boltzmann's 3D transport equation is known as the Kramers-Moyal expansion.

The equation for Gaussian diffusion is a consequence of Fick's first law [26] which states that the diffusive flux is a linear function of the concentration gradient. The flux equation can be expanded to accommodate the more general model of diffusion described by the Kramers-Moyal expansion. Each subsequent term involves a higher rank tensor - a moment of the displacement probability density function (pdf) - and higher order derivatives of the concentration. Applying Fick's second law (mass conservation) to the generalized flux we obtain the so-called Kramers-Moyal expansion of the diffusion equation,

$$\frac{\partial u}{\partial t} = \mathrm{div}(D_{ij}\nabla_j u + D_{ijk}\nabla_{jk}u + D_{ijkl}\nabla_{jkl}u...). \tag{12}$$

The nabla symbol (∇) with n subscripts is a compact notation denoting a tensor of rank n whose components are partial derivatives of degree n,

$$\nabla_{i_1 i_2 \ldots i_n} = \frac{\partial^n}{\partial x_{i_1} \partial x_{i_2} \ldots \partial x_{i_n}}. \tag{13}$$

In the next section we will describe in more detail the tensor notation used in equation (12), and describe how symmetry constaints on the tensors make the simulation of generalized diffusion more efficient. Later, we will show how to estimate the diffusion displacement pdf from the tensors that characterize the moments of the pdf.

3 High-Rank Tensors

The *rank* of the tensor (referred to as order in some literature) is the number of indices into it. Tensors of rank 0 are scalars, and tensors of rank 1 are vectors. A rank-2 tensor can be represented as a matrix. If d is the *dimension* of the tensor, then each index can take one of d different values. In 2 dimensions a rank-ℓ tensor then has 2^ℓ components. A general tensor may also have two different types of indices, covariant and contravariant, usually denoted using subscripts and superscripts. For Cartesian tensors these are equivalent, so will denote indices using only subscripts.

High-rank tensors have recently been applied to describe diffusion in the context of medical imaging by Özarslan et al. [27,28] and Liu et al. [29]. Diffusion within biological structures can be non-Gaussian, for example, when bifurcating or intersecting white-matter fiber bundles occur in the brain.

Imposing a symmetry constraint on D will reduce the number of unique tensor components and eliminate advective transport from the equation. For symmetric rank-2 tensors we have

$$D_{ij} = D_{ji}. \tag{14}$$

In general, symmetry implies that components whose indices are permutations of each other are equal,

$$D_{i_1 i_2 \ldots i_\ell} = D_{(i_1 i_2 \ldots i_\ell)} \tag{15}$$

where $(i_1 i_2 \ldots i_\ell)$ stands for all permutations of the ℓ indices. For a symmetric tensor, a component can be identified by the number of times each index occurs, order notwithstanding. As a result of symmetry the number of unique components of a symmetric 2-dimensional tensor is much lower than 2^ℓ. In fact, the number of unique elements is given by $N_\ell = \ell + 1$.

The number of permutations of each set of indices is the same as the number of times that component is duplicated. This number is called the multiplicity, and is given by

$$\mu = \frac{\ell!}{n_x! n_y!} \tag{16}$$

where n_x, n_y are the number of x and y indices respectively.

In writing an expression containing tensors, we will use the summation convention. This means that repeated indices are to be multiplied pairwise, and summed over all possible values,

$$A_{i_1 i_2 \ldots i_\ell} B_{i_1 i_2 \ldots i_\ell} = \sum_{i_1=1}^{d} \sum_{i_2=1}^{d} \ldots \sum_{i_\ell=1}^{d} A_{i_1 i_2 \ldots i_\ell} B_{i_1 i_2 \ldots i_\ell}. \tag{17}$$

If both tensors are symmetric, the number of terms in the summation can be greatly reduced by only summing over the unique components

$$A_{i_1 i_2 \ldots i_\ell} B_{i_1 i_2 \ldots i_\ell} = \sum_{k=1}^{N_\ell} \mu_k A_k B_k \tag{18}$$

where A_k, B_k are the k-th unique components of A and B, and μ_k is the multiplicity of that component.

Truncating the Kramers-Moyal expansion (12) at the 4th term, for homogeneous diffusion we arrive at

$$\frac{\partial u}{\partial t} = D_{ij} \nabla_{ij} u + D_{ijk} \nabla_{ijk} u + D_{ijkl} \nabla_{ijkl} u. \tag{19}$$

For symmetric tensors D, the expansion (19) has only 12 terms,

$$\begin{aligned} \frac{\partial u}{\partial t} = & D_{xx} \partial_{xx} u + 2 D_{xy} \partial_{xy} u + D_{yy} \partial_{yy} u && (20) \\ & + D_{xxx} \partial_{xxx} u + 3 D_{xxy} \partial_{xxy} u + 3 D_{xyy} \partial_{xyy} u + D_{yyy} \partial_{yyy} u \\ & + D_{xxxx} \partial_{xxxx} u + 4 D_{xxxy} \partial_{xxxy} u + 6 D_{xxyy} \partial_{xxyy} u \\ & + 4 D_{xyyy} \partial_{xyyy} u + D_{yyyy} \partial_{yyyy} u. && (21) \end{aligned}$$

	isotropic	anisotropic Gaussian	anisotropic non-Gaussian
hom.	$d\nabla^2 u$	$\mathrm{div}(D\nabla u)$	$D_{ij}\nabla_{ij}u + D_{ijk}\nabla_{ijk}u + D_{ijkl}\nabla_{ijkl}u$
inhom.	$d\nabla^2 u+$ $\nabla u \cdot \nabla d$	$tr(DH(u))+$ $\nabla u \cdot \mathrm{div}\, D$	$D_{ij}\nabla_{ij}u + D_{ijk}\nabla_{ijk}u + D_{ijkl}\nabla_{ijkl}u+$ $\nabla_i D_{ij}\nabla_j u + \nabla_i D_{ijk}\nabla_{jk}u + \nabla_i D_{ijkl}\nabla_{jkl}u$

Table 1
Diffusion term for all cases of homogeneous/inhomogeneous, isotropic/anisotropic Gaussian/anisotropic non-Gaussian.

In Table (1) we summarize the various forms which the diffusion term may take. In these equations d is a scalar diffusion coefficient, D is a diffusion tensor, $H(u)$ is the Hessian of u, $tr()$ denotes the trace and

$$\mathrm{div}\, D = \begin{bmatrix} \mathrm{div}[D_{xx} D_{xy}]^T \\ \mathrm{div}[D_{yx} D_{yy}]^T \end{bmatrix}. \tag{22}$$

In general inhomogeneous diffusion leads to an equation with many more terms than homogeneous diffusion. Also note that in Table (1) the isotropic and anisotropic Gaussian diffusion equations are special cases of the more general non-Gaussian diffusion equation.

In this work we have discussed the generation of textures on 2-dimensional Euclidean domains, but the same equations can be used to generate 3-dimensional texture volumes by extending the noise texture to 3 dimensions and using the 3D equation for the gradient and divergence. The problem of computing generalized reaction-diffusion on manifolds is not considered here, but can be incorporated into this framework. Using the intrinsic divergence and gradient operators for the manifold, which will depend on the metric tensor of the surface, will permit the process to be accurately simulated on curved domains. This was the approach taken by Diewald et al. [21] for Gaussian diffusion on manifolds.

4 Gram-Charlier Expansion of the Diffusion Displacement PDF

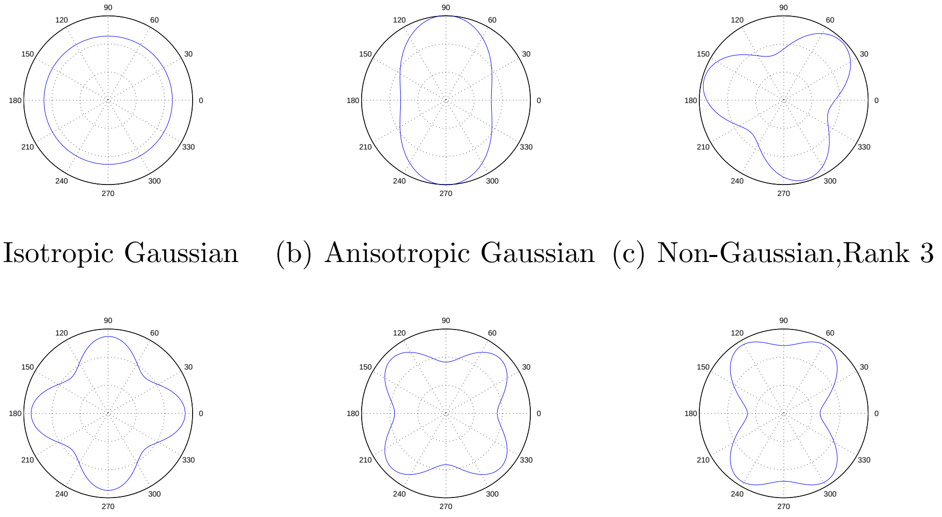
The Gram-Charlier expansion [30] of the normal distribution is popular in the field of mathematical finance because covariance, skewness and kurtosis can be specified as parameters. For multivariate distributions the moments are tensors : the mean is a rank-1 tensor (or vector), variance is a rank-2 tensor (or matrix), skewness is a rank-3 tensor, and kurtosis is a rank-4 tensor. These tensors appeared in the Kramer-Moyals expansion presented earlier. The diffusion propagator for the generalized diffusion equation (12) has Gram-Charlier expansion given by

$$p_\tau(\mathbf{r}) = N(0, 2\tau D^{(2)})(1 + \frac{Q^{(3)}}{3!}H^{(3)}(\mathbf{r}) + \frac{Q^{(4)}}{4!}H^{(4)}(\mathbf{r})...), \tag{23}$$

where the cumulants of $p_\tau(\mathbf{r})$ are given by

$$Q^{(n)} = (-1)^n n! D^{(n)} \tau, \tag{24}$$

$D^{(n)}$ denotes the tensor of rank n and $H^{(n)}(\mathbf{r})$ are the Hermite tensor polynomials (see appendix). Since the Hermite polynomials are orthogonal with respect to the Gaussian function, the expansion (23) can be shown to integrate to one. However, for some parameter values, this expansion may produce small negative values in the tails.



(a) Isotropic Gaussian (b) Anisotropic Gaussian (c) Non-Gaussian,Rank 3

(d) Non-Gaussian,Rank 4 (e) Non-Gaussian,Rank 4 (f) Non-Gaussian,Rank 4

Fig. 1. Polar plots of the diffusion propagators, $p_\tau(\mathbf{r})$ for $|\mathbf{r}| = 1$ and $\tau = 0.1$ sec.

The peaks of $p_\tau(\mathbf{r})$ correspond to the dominant orientations of the resulting texture. Plots of 6 different diffusion propagators are shown in Figure (1) for $\tau = 0.1, |\mathbf{r}| = 1$. The corresponding tensor components are shown in Table (2). These correspond to the tensor sequences used to produce the images in the Results section.

5 Implementation

The system of reaction-diffusion equations can be linearized and written in either explicit or semi-implicit form. The explicit formulation requires a small

Tensor components	(a)	(b)	(c)	(d)	(e)	(f)
D_{xx}	1	1	1	1	1	1
D_{xy}	0	0	0	0	0	0
D_{yy}	1	2	1	1	1	2
D_{xxx}	0	0	1	0	0	0
D_{xxy}	0	0	-1	0	0	0
D_{xyy}	0	0	-1	0	0	0
D_{yyy}	0	0	1	0	0	0
D_{xxxx}	0	0	0	1	-1	-1
D_{xxxy}	0	0	0	0	0	0
D_{xxyy}	0	0	0	-1	1	1
D_{xyyy}	0	0	0	0	0	0
D_{yyyy}	0	0	0	1	-1	-1

Table 2
Tensor components for the sequences producing the propagators in Figure 1.

step size for stability and each iteration can be performed quite quickly. The semi-implicit formulation remains stable for larger step sizes, but requires the solution of a sparse linear system for each iteration.

Using central differences the Laplacian $\nabla^2 u$ can be discretized as the convolution $K * u$ and implemented as $A\mathbf{u}$ where

$$K = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad A = \begin{bmatrix} -4 & 1 & 0 & \cdots & 1 & 0 & \cdots & 0 \\ 1 & -4 & 1 & 0 & \cdots & 1 & 0 & \cdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & 0 & \cdots & 1 & \cdots & 0 & 1 & -4 \end{bmatrix} \tag{25}$$

and u is reshaped into a column vector, $\mathbf{u}$.

The explicit formulation of isotropic reaction-diffusion in terms of the matrix A is given by

$$\frac{\mathbf{u}^{t+1} - \mathbf{u}^t}{\delta} = A\mathbf{u}^t + f(\mathbf{u}^t). \tag{26}$$

where δ is the time-step and superscripts denote iteration number. The corresponding update equation for $\mathbf{u}$ is

$$\mathbf{u}^{t+1} = (I + \delta A)\mathbf{u}^t + \delta f(\mathbf{u}^t). \tag{27}$$

A semi-implicit formulation can be obtained by time-lagging the reaction term resulting in the update equation

$$\mathbf{u}^{t+1} = (I - \delta A)^{-1}(\mathbf{u}^t + \delta f(\mathbf{u}^t)). \tag{28}$$

In both cases, the matrix A is symmetric and sparse, with 5 nonzero diagonals. For tensor anisotropic diffusion characterized by the rank-2 tensor D_{ij}, $\text{div}(D\nabla u) = K * u$ where the convolution kernel is

$$K = \frac{1}{2}\begin{bmatrix} -D_{xy} & 2D_{yy} & D_{xy} \\ 2D_{xx} & -4(D_{xx} + D_{yy}) & 2D_{xx} \\ D_{xy} & 2D_{yy} & -D_{xy} \end{bmatrix}. \tag{29}$$

In this case the corresponding matrix, A, will have 9 nonzero diagonals. The non-Gaussian diffusion operator described in the next section will be written as a 5×5 convolution kernel, allowing the generalized reaction-diffusion equation to be computed using either update formula (27) or (28). Periodic boundary conditions allow the texture to seamlessly tile, but at the expense of 4 additional diagonals in the matrix A. If the diffusion is homogeneous, the matrix will remain symmetric.

Multigrid algorithms work by approximating the solution to a problem on a coarse grid, then progressively refining the solution using finer grids. Witkin and Kass [9] describe such an implementation for anisotropic reaction-diffusion textures, and it is straightforward to implement our technique in such a framework.

5.1 Savitsky-Golay Filters

Numerical derivatives of discretized functions are sensitive to noise in the data. One solution to this problem is to fit a low-degree polynomial to the region surrounding each pixel, then differentiate that polynomial. Since the polynomial is so simple to differentiate, the complexity of these schemes is dominated by the fitting process. The technique presented by Savitsky and Golay [31] involves the least-squares fitting of a polynomial to the neighborhood of each pixel. Since much of the fitting procedure can be precomputed, the technique is very fast.

Consider the 5×5 neighborhood of each pixel (r_0, s_0). The local coordinates $(x, y) = (r - r_0, s - s_0)$ for the polynomial surface will have the origin at the location of pixel (r_0, s_0), as shown

$$\begin{bmatrix} (-2,-2) & (-1,-2) & (0,-2) & (+1,-2) & (+2,-2) \\ (-2,-1) & (-1,-1) & (0,-1) & (+1,-1) & (+2,-1) \\ (-2,0) & (-1,0) & (0,0) & (+1,0) & (+2,0) \\ (-2,+1) & (-1,+1) & (0,+1) & (+1,+1) & (+2,+1) \\ (-2,+2) & (-1,+2) & (0,+2) & (+1,+2) & (+2,+2) \end{bmatrix}. \tag{30}$$

A degree n polynomial $p_{rs}(x, y)$ will be fit to the image intensities in this neighborhood,

$$\begin{aligned} p_{rs}(x,y) &= a_{00} + a_{10}x + a_{01}y + a_{20}x^2 + a_{11}xy + a_{02}y^2 ... \quad (31) \\ &= \sum_{i=0}^{n} \sum_{j=0}^{n} a_{ij} x^i y^j. \quad (32) \end{aligned}$$

The fitting is performed by solving, in the least-squares sense, the system of equations

$$\begin{bmatrix} 1 & x_1 & y_1 & x_1^2 & y_1^2 & x_1 y_1 & \cdots \\ 1 & x_2 & y_1 & x_2^2 & y_1^2 & x_2 y_1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_m & y_m & x_m^2 & y_m^2 & x_m y_m & \cdots \end{bmatrix} \begin{bmatrix} a_{00} \\ a_{10} \\ a_{01} \\ a_{20} \\ a_{02} \\ a_{11} \\ \vdots \\ a_{nn} \end{bmatrix} = \begin{bmatrix} u(x_1 + r_0, y_1 + s_0) \\ u(x_2 + r_0, y_1 + s_0) \\ \vdots \\ u(x_m + r_0, y_m + s_0) \end{bmatrix}. \tag{33}$$

The overconstrained system (33) is solved for the polynomial coefficients, $\mathbf{a}$. Consider the linear system to be $X\mathbf{a} = \mathbf{u}$. Then the least squares solution can be written in terms of the pseudoinverse as

$$\mathbf{a} = (X^T X)^{-1} X^T \mathbf{u} = X^+ \mathbf{u}. \tag{34}$$

Note that the matrix X^+ does not depend on the data, $\mathbf{u}$, and can be precomputed for a fixed neighborhood size and polynomial degree. Likewise, derivatives of the polynomial $p_{rs}(x, y)$ depend on the coefficients a, and can be written as filter convolution kernels. For more details about these filters see the original paper by Savitsky and Golay [31] or the numerical methods book by Press et al. [32].

The partial derivatives at image coordinates (r_0, s_0) can then be approximated by partial derivatives of $p_{rs}(x = 0, y = 0)$. The fourth and third order partial derivatives are

$$\begin{aligned}
&\partial_{xxxx} p_{rs}(0,0) = 24a_{40}, \quad \partial_{xxx} p_{rs}(0,0) = 6a_{30}, \\
&\partial_{xxxy} p_{rs}(0,0) = 6a_{31}, \quad \partial_{xxy} p_{rs}(0,0) = 2a_{21}, \\
&\partial_{xxyy} p_{rs}(0,0) = 4a_{22}, \quad \partial_{xyy} p_{rs}(0,0) = 2a_{12}, \\
&\partial_{xyyy} p_{rs}(0,0) = 6a_{13}, \quad \partial_{yyy} p_{rs}(0,0) = 6a_{03}, \\
&\partial_{yyyy} p_{rs}(0,0) = 24a_{04}.
\end{aligned} \tag{35}$$

The generalized diffusion equation can be rewritten in terms of the coefficients as

$$\begin{aligned}
\frac{\partial u}{\partial t} =& 2(D_{xx}a_{20} + D_{xy}a_{11} + D_{yy}a_{02}) \\
&+ 6(D_{xxx}a_{30} + D_{xxy}a_{21} + D_{xyy}a_{12} + D_{yyy}a_{03}) \\
&+ 24(D_{xxxx}a_{40} + D_{xxxy}a_{31} + D_{xxyy}a_{22} + D_{xyyy}a_{13} + D_{yyyy}a_{04}).
\end{aligned} \tag{36}$$

We can discretize Equation (19), and rewrite as Equations (27) and (28) by letting

$$\begin{aligned}
A =& 2(D_{xx}X^+_{20} + D_{xy}X^+_{11} + D_{yy}X^+_{02}) \\
&+ 6(D_{xxx}X^+_{30} + D_{xxy}X^+_{21} + D_{xyy}X^+_{12} + D_{yyy}X^+_{03}) \\
&+ 24(D_{xxxx}X^+_{40} + D_{xxxy}X^+_{31} + D_{xxyy}X^+_{22} + D_{xyyy}X^+_{13} + D_{yyyy}X^+_{04})
\end{aligned} \tag{37}$$

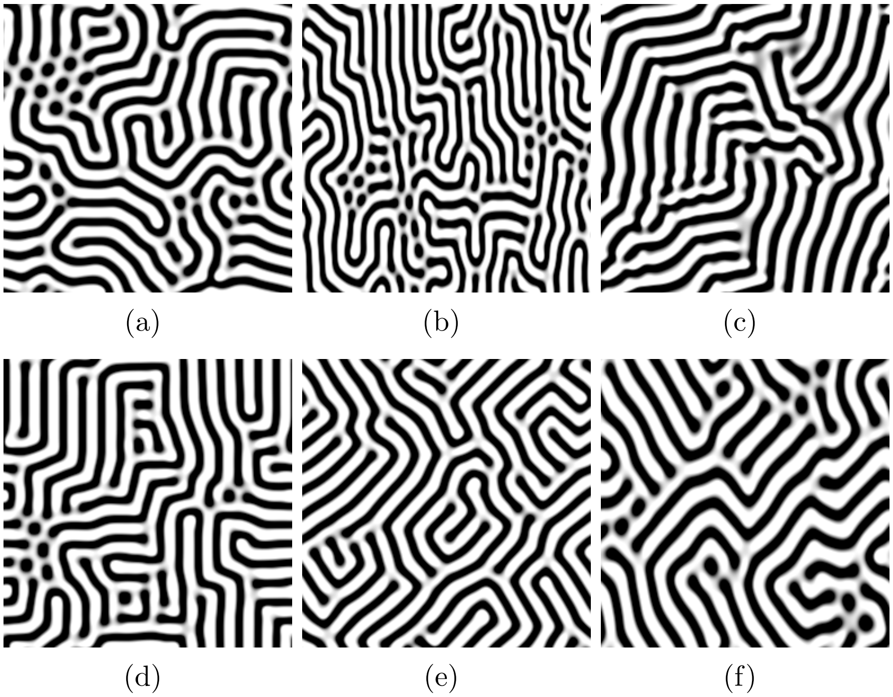
where X^+_{ij} is the row of X^+ such that $X^+_{ij}u = a_{ij}$. The result is a matrix which has 25 nonzero diagonals.

The numerical stability of the explicit simulation depends on the spectral radius, ρ, of the matrix $(I + \delta A)$. For values greater than one, small errors can be amplified upon repeated matrix-vector multiplication. As constructed by equation (37) we compute $\rho(I + \delta A) \approx 1.102$ for the tensor sequences in table 2 with $\delta = 0.1$. By replacing the Savitsky-Golay second partial derivative approximation kernels $X^+_{20}, X^+_{11}, X^+_{02}$ with standard second-order central difference kernels, the spectral radius can be improved. For our experiments

$\rho(I + \delta A) \approx 1.03$ and concentration values remained bounded during simulation. Likewise, the condition number of the matrix $(I - \delta A)$ which affects the numerical stability of the semi-implicit simulation can be improved by this substitution.

6 Results

The generalized diffusion model was used in conjunction with the Turing reaction model. Initial conditions were $u = 4 + U(-2, 2)$, $v = 4 + U(-2, 2)$ where $U(min, max)$ is a uniformly distributed random value between min and max at each pixel. The parameters values were $\delta = 0.1, \alpha = 12, \beta = 15$. Degree 5 Savitzky-Golay filters were used to compute derivatives of order higher than 2. The results generated using the tensor sequences in Table (2) and Figure (1) are shown in Figure (2). A multigrid scheme was used for all results in this section. A total of 3 grid discretizations $(64 \times 64, 128 \times 128, 256 \times 256)$ were used, and 20000 iterations at each grid level were performed using the explicit scheme. Note that in each case the dominant orientations in each texture correspond to the peaks of the diffusion displacement function plotted in Figure (1). In each case, increasing the rank of the tensor expansion increases the number of peaks in the diffusion propagator. Expansions with only even

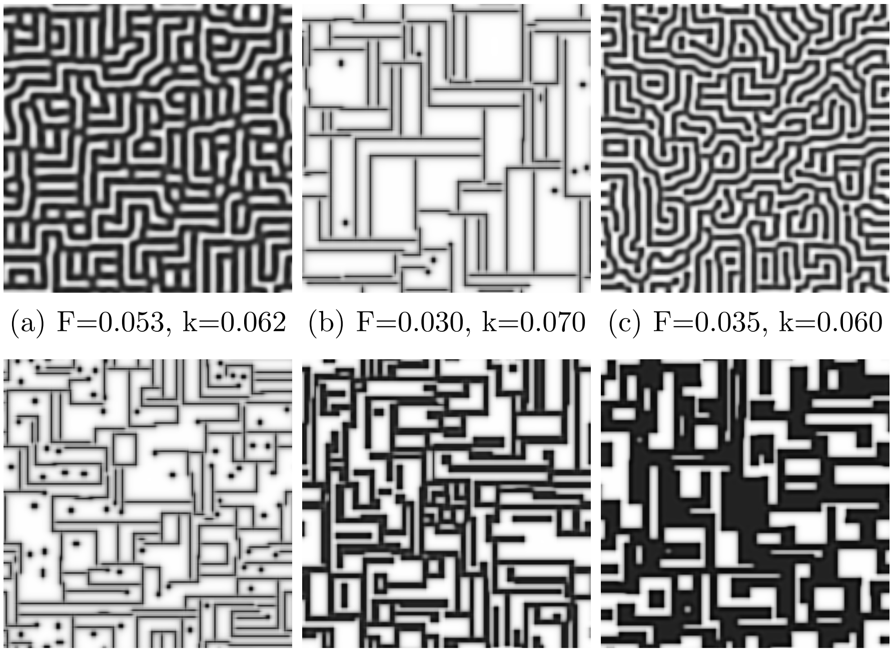

(a) (b) (c)

(d) (e) (f)

Fig. 2. Textures generated by expansion of Turing reaction-diffusion equation with $\alpha = 12$, $\beta = 15$. The tensor sequences used correspond to those in Figure(1) and Table(2).

order terms are antipodally symmetric, so expansions (d), (e) and (f) lead to patterns with 2 dominant texture orientations. However, the rank 3 expansion (c) has 3 dominant orientations.

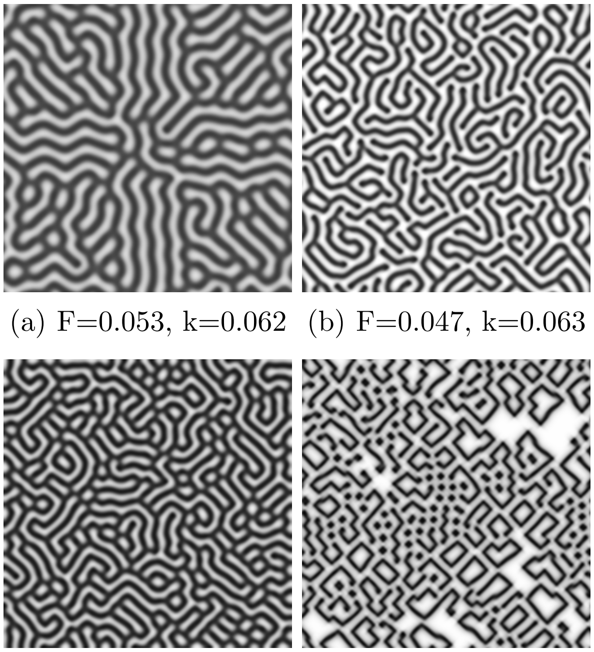
The generalized diffusion model was also used in conjunction with the Gray-Scott reaction model. Initial conditions were $u = 1, v = 0$, then perturbed by setting a square region in the center of u to value 0.5, and the same sized square region in v to 0.25. Both u and v then each had uniform random noise $U(-0.25, 0.25)$ added. The timestep was set to $\delta = 0.1$. The 4th order expansion (d) was used to generate the patterns in Figure (3). The preferred orientations are axis-aligned in this case. The 4th order expansion (e) was used to generate the patterns in Figure (4). The preferred orientations are diagonal for these patterns. The 3rd order expansion (c) was used to generate the patterns in Figure (5). Compared to the Turing model, the Gray-Scott model produced a more varied set of textures as the parameters were varied. In all cases we were able to control the texture orientation in a predictable way by modifying the tensor expansion. The diffusion process was homogeneous for all of our results - the same tensor sequence was used at each pixel for u and v. The orientational heterogeneity in these patterns is due to the non-Gaussian diffusion model and the random initial conditions. Other patterns can be obtained by choosing different tensor expansions for each of the morphogens.


(a) F=0.053, k=0.062 (b) F=0.030, k=0.070 (c) F=0.035, k=0.060

(d) F=0.035, k=0.070 (e) F=0.060, k=0.062 (f) F=0.065, k=0.060

Fig. 3. Textures generated by rank 4 expansion (d) of Gray-Scott reaction-diffusion equation.

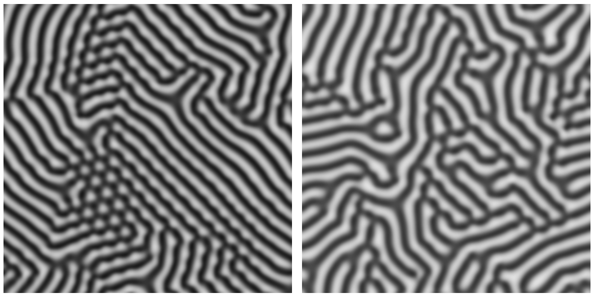
Figure (6) was generated by allowing the Gray-Scott reaction parameters to



(a) F=0.053, k=0.062 (b) F=0.047, k=0.063

(c) F=0.040, k=0.060 (d) F=0.040, k=0.064

Fig. 4. Textures generated by rank 4 expansion (e) of Gray-Scott reaction-diffusion equation.



(a) F=0.039, k=0.059 (b) F=0.044, k=0.061

Fig. 5. Textures generated by rank 3 expansion (c) of Gray-Scott reaction-diffusion equation.

vary spatially. This parameter map can be used to determine the nature of the textures generated from various parameter values. For example, it can be seen that $F = 0.06, k = 0.03$ results in a constant valued texture, as does $F = 0.06, k = 0.065$. The interesting textures appear along the border between two regions characterized by trivial steady states. Such maps can eliminate the trial and error associated with picking parameter values. Normal maps computed from generalized reaction-diffusion textures were used to render the models in Figure (7). In contrast to previous reaction-diffusion techniques, the angular features and corners in these textures suggest man-made structures, such as etched pottery, riveted panels forming the surface of a vehicle, quilted fabrics or embossed metals. The periodic boundary condition imposed during

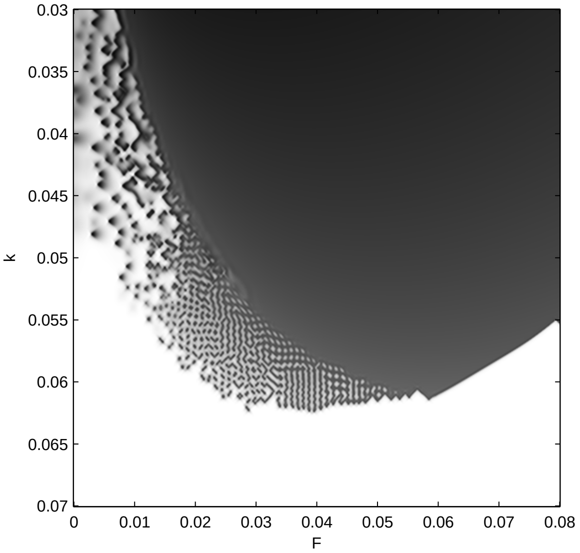


Fig. 6. Gray-Scott parameter map for tensor expansion (e).

texture generation allows these maps to be tiled multiple times over the surface without seams appearing.

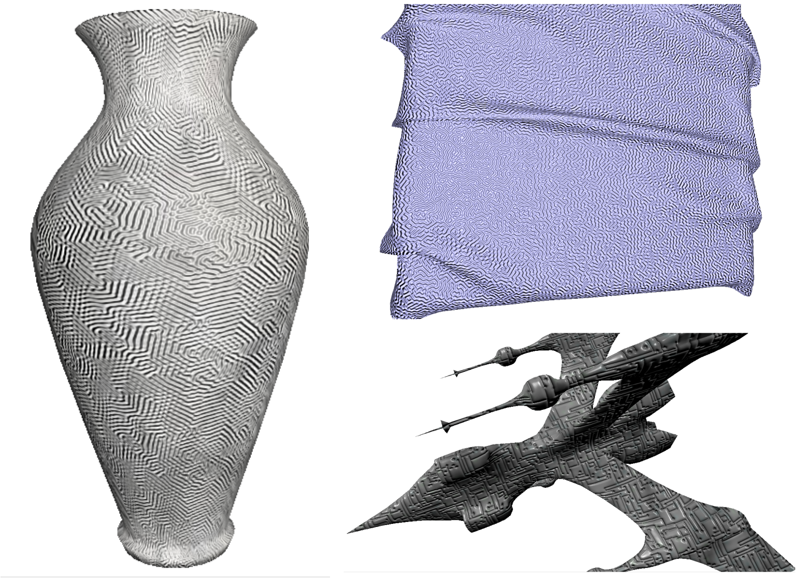


Fig. 7. Generalized reaction-diffusion textures as bump maps

Timing results are given in Table (3) for a GPU implementation written in OpenGL shading language and tested on an Nvidia 8800 GTX with 768MB RAM. The isotropic case was implemented using convolution by the 3×3 kernel in Equation (25) and the anisotropic Gaussian case was implemented using the kernel given in Equation (29). The non-Gaussian case was implemented using 5×5 convolution with the kernel computed as described in the previous section. In all cases the kernel values were precomputed on the CPU and uploaded to the GPU.

	isotropic	anisotropic	anisotropic
size		Gaussian	non-Gaussian
256×256	6387	6245	6204
512×512	6206	3740	2031
1024×1024	1515	1210	603

Table 3
Iterations per second for the GPU implementation.

This implementation makes it possible to generate 512×512 generalized reaction-diffusion textures in about 15 seconds using a multigrid scheme with 20000 iterations at each resolution.

7 Conclusion

Generalizing the model of diffusion used in reaction-diffusion texture synthesis makes many new textures possible. By simulating a non-Gaussian diffusion process, textures with orientational variability are possible, even when the diffusion is homogeneous. Inhomogeneous diffusion requires the specification of a tensor field, and the evaluation of additional terms in the diffusion equation due to the spatially varying diffusion coefficients. Generalized homogeneous diffusion, on the other hand, requires the specification of only 9 additional parameters for an order 4 expansion. Previous schemes have employed multiple coupled systems of equations to produce patterns with competing orientations, but our technique can produce such pattern using only a single system (one pair of morphogens).

The resulting patterns are strikingly inorganic. They feature multiple orientations and sharp corners. Some resemble building footprints, circuit boards, mazes, and networks of intersecting roadways. Since the character of the textures depends strongly on the reaction terms of the system of equations, this new diffusion model can be used to extend previously described reaction-diffusion systems.

Numerical simulation of diffusion has many applications, including image processing, scientific visualization and medical imaging. Our future work will involve using an inhomogeneous generalized diffusion model for controlling exemplar-based texture synthesis, image filtering, visualizing fields of probability density functions and also visualizing uncertainty.

Appendix

The Hermite tensor polynomials, up to order 4, are given by

$$\begin{aligned}
H^{(0)} &= 1 \\
H_i^{(1)} &= x_i \\
H_{i,j}^{(2)} &= x_i x_j - \delta_{ij} \\
H_{i,j,k}^{(3)} &= x_i x_j x_k - (x_i \delta_{jk} + x_j \delta_{ik} + x_k \delta_{ij}) \\
H_{i,j,k,l}^{(4)} &= x_i x_j x_k x_l - (\delta_{ij} x_k x_l + \delta_{ik} x_j x_l + \delta_{il} x_j x_k \\
&\quad + \delta_{jk} x_i x_l + \delta_{jl} x_i x_k + \delta_{kl} x_i x_j) - 3\delta_{ijkl}.
\end{aligned} \tag{38}$$

where

$$\delta_{i_1 i_2 \ldots i_l} = \begin{cases} 1 \text{ for } i_1 = i_2 \ldots = i_l \\ 0 \text{ otherwise} \end{cases} \tag{39}$$

Higher order polynomials can be generated using the recurrence relation given by Grad [33].

Acknowledgements

Thanks to Baba Vemuri and Evren Özarslan for enlightening discussions. This research was funded by WVU start-up funding.

References

[1] D. S. Ebert, F. K. Musgrave, D. Peachey, K. Perlin, S. Worley, Texturing and Modeling: A Procedural Approach, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002.

[2] K. Perlin, An image synthesizer, in: Computer Graphics (Proceedings of ACM SIGGRAPH 85), ACM Press, New York, NY, USA, 1985, pp. 287–296.

[3] K. Perlin, E. M. Hoffert, Hypertexture, in: Computer Graphics (Proceedings of ACM SIGGRAPH 89), ACM Press, New York, NY, USA, 1989, pp. 253–262.

[4] S. Worley, A cellular texture basis function, in: Proceedings of ACM SIGGRAPH 96, ACM Press / ACM SIGGRAPH, New York, NY, USA, 1996, pp. 291–294.

[5] J. F. Blinn, M. E. Newell, Texture and reflection in computer generated images, Commun. ACM 19 (10) (1976) 542–547.

[6] K. W. Fleischer, D. H. Laidlaw, B. L. Currin, A. H. Barr, Cellular texture generation, in: Proceedings of ACM SIGGRAPH 95, ACM Press / ACM SIGGRAPH, New York, NY, USA, 1995, pp. 239–248.

[7] M. Ashikhmin, Synthesizing natural textures, in: SI3D '01: Proceedings of the 2001 symposium on Interactive 3D graphics, ACM Press, New York, NY, USA, 2001, pp. 217–226.

[8] A. M. Turing, The chemical basis of morphogenesis, Philosophical Transactions of the Royal Society (B) 237 (1952) 37–72.

[9] A. Witkin, M. Kass, Reaction-diffusion textures, in: Proceedings of the 18th annual conference on Computer graphics and interactive techniques, 1991, pp. 299–308.

[10] G. Turk, Generating textures on arbitrary surfaces using reaction-diffusion, in: Proceedings of the 18th annual conference on Computer graphics and interactive techniques, 1991, pp. 289–298.

[11] D. R. Fowler, H. Meinhardt, P. Prusinkiewicz, Modeling seashells, in: Computer Graphics (Proceeding of ACM SIGGRAPH 92), ACM Press, New York, NY, USA, 1992, pp. 379–387.

[12] J. D. Murray, Mathematical Biology I, Springer, New York, NY, USA, 2002.

[13] J. D. Murray, M. R. Myerscough, Pigmentation pattern formation on snakes., J Theor Biol. 149 (3) (1991) 339–360.

[14] S. Kondo, R. Asai, A reaction-diffusion wave on the skin of the marine angelfish pomacanthus., Nature 376 (1995) 765–768.

[15] A. Gierer, H. Meinhardt, Theory of Biological Pattern Formation, Kybernetik 12 (1972) 30–39.

[16] M. J. Harris, G. Coombe, T. Scheuermann, A. Lastra, Physically-based visual simulation on graphics hardware, in: HWWS '02: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2002, pp. 109–118.

[17] A. Sanderson, M. Kirby, C. Johnson, L. Yang, Advanced reaction-diffusion models for texture synthesis, Journal of Graphics Tools 11 (3) (2006) 47–71.

[18] P. Gray, S. Scott, Sustained oscillations and other exotic patterns of behaviour in isothermal reactions, J. Physical Chemistry 59 (1985) 22–32.

[19] J. E. Pearson, Complex patterns in a simple system, Science 261 (5118).

[20] G. K. Vallis, Atmospheric and Oceanic Fluid Dynamics: Fundamentals and Large-Scale Circulation, Cambridge University Press, Cambridge, U.K., 2006.

[21] U. Diewald, T. Preußer, M. Rumpf, Anisotropic diffusion in vector field visualization on Euclidean domains and surfaces, IEEE Transactions on Visualization and Computer Graphics 6 (2) (2000) 139–149.

[22] G. Kindlmann, D. Weinstein, D. Hart, Strategies for direct volume rendering of diffusion tensor fields, IEEE Transactions on Visualization and Computer Graphics 6 (2) (2000) 124–138.

[23] J. Weickert, A review of nonlinear diffusion filtering, in: Proceedings of the First International Conference on Scale-Space Theory in Computer Vision, 1997, pp. 3–28.

[24] P. Oliva, D. H. Zanette, One-species bimolecular reaction kinetics enhanced by anomalous diffusion, Physical Review E 54 (2) (1996) 1366–1368.

[25] A. Einstein, On the movement of small particles suspended in stationary liquids required by the molecular-kinetic theory of heat, in: The Collected Papers of Albert Einstein: Volume 2: The Swiss Years: Writings, 1900–1909. (English translation), Princeton University Press, Princeton, 1989, pp. 123–134, transl. by Anna Beck and Peter Havas.

[26] D. Wilkinson, Mass Transport in Solids and Fluids, Cambridge University Press, Cambridge, U.K., 2000.

[27] E. Özarslan, T. H. Mareci, Generalized diffusion tensor imaging and analytical relationships between diffusion tensor imaging and high angular resolution diffusion imaging, Magn Reson Med 50 (2003) 955–965.

[28] E. Özarslan, B. C. Vemuri, T. H. Mareci, Generalized scalar measures for diffusion MRI using trace, variance and entropy, Magn Reson Med 53 (4) (2005) 866–876.

[29] C. Liu, R. Bammer, B. Acar, M. E. Moseley, Characterizing non-gaussian diffusion by using generalized diffusion tensors, Magnetic Resonance in Medicine 51 (2004) 924–937.

[30] C. Rose, M. D. Smith, Mathematical Statistics with Mathematica, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2002.

[31] A. Savitsky, M. J. E. Golay, Smoothing and differentiation of data by simplified least squares procedures, Anal. Chem. 36 (1964) 1627–1639.

[32] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, Numerical Recipes in C: The Art of Scientific Computing, Cambridge University Press, New York, NY, USA, 1992.

[33] H. Grad, Note on n-dimensional hermite polynomials, Commun. Pure Appl. Math. 2 (1949) 325–330.

[34] G. Turk, Texture synthesis on surfaces, in: SIGGRAPH ’01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques, ACM Press, New York, NY, USA, 2001, pp. 347–354.

[35] S. Lefebvre, H. Hoppe, Parallel controllable texture synthesis, in: SIGGRAPH '05: ACM SIGGRAPH 2005 Papers, ACM Press, New York, NY, USA, 2005, pp. 777–786.

[36] S. Lefebvre, H. Hoppe, Appearance-space texture synthesis, in: SIGGRAPH '06: ACM SIGGRAPH 2006 Papers, ACM Press, New York, NY, USA, 2006, pp. 541–548.

[37] T. Preußer, M. Rumpf, Anisotropic nonlinear diffusion in flow visualization, in: IEEE Visualization, 1999, pp. 325–332.

[38] D. Tschumperle, R. Deriche, Tensor field visualization with PDEs and application to DT-MRI fiber visualization, in: Proc. of Variational and Level Set Methods, 2003.

[39] A. R. Sanderson, C. R. Johnson, R. M. Kirby, Display of vector fields using a reaction-diffusion model, in: VIS '04: Proceedings of the conference on Visualization '04, IEEE Computer Society, Washington, DC, USA, 2004, pp. 115–122.