Physically-based Surface Texture Synthesis Using a Coupled Finite Element System

Chandrajit Bajaj[1], Yongjie Zhang[2], and Guoliang Xu[3]

[1] Department of Computer Sciences and Institute for Computational Engineering and Sciences, The University of Texas at Austin, USA. bajaj@cs.utexas.edu

[2] Department of Mechanical Engineering, Carnegie Mellon University, USA. jessicaz@andrew.cmu.edu

[3] Institute of Computational Mathematics, Academy of Mathematics and System Sciences, Chinese Academy of Sciences, China. xuguo@lsec.cc.ac.cn

Abstract. This paper describes a stable and robust finite element solver for physically-based texture synthesis over arbitrary manifold surfaces. Our approach solves the reaction-diffusion equation coupled with an anisotropic diffusion equation over surfaces, using a Galerkin based finite element method (FEM). This method avoids distortions and discontinuities often caused by traditional texture mapping techniques, especially for arbitrary manifold surfaces. Several varieties of textures are obtained by selecting different values of control parameters in the governing differential equations, and furthermore enhanced quality textures are generated by fairing out noise in input surface meshes.

Key words: texture synthesis, finite element method, reaction-diffusion equation, anisotropic diffusion.

1 Introduction

Texture mapping is an essential technique to map textures onto computer models, but it usually introduces distortions or discontinuities. An alternate approach is to synthesize a texture directly on the surface of objects. The finite difference method (FDM) has been extended to generate patterns over surfaces, but the problem of distortions or discontinuities still exist, and the results can be often divergent if non-suitable time steps or incorrect initial conditions are chosen. Here we use the finite element method (FEM) for the stable and robust solution of reaction-diffusion partial differential equations (PDEs). These FEM solutions allow for smooth and distortion free textures to be directly synthesized and visualized on arbitrary surface manifolds.

However when the input surface meshes are noisy, the textures synthesized by FEM solutions are often equally noisy, and as shown in Figure 1. To correct this, we couple an anisotropic diffusion PDE with the reaction-diffusion PDE's to remove noise on the surface while preserving geometric features, and synthesizing smooth textures at the same time. Enhanced and distortion-free textures are generated by using this coupled PDE system.

The main steps of our finite element texture synthesis approach are as follows:

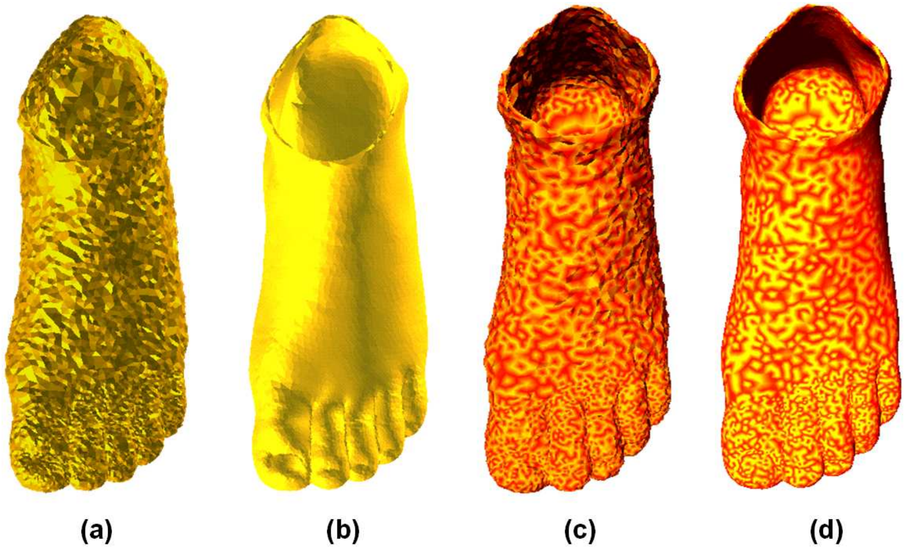

Fig. 1. Physically-based textures synthesized on the manifold foot surface. (a) - the noisy surface; (b) - anisotropic diffusion is applied on (a) to remove noise while preserving geometric features; (c) - reaction-diffusion is applied on (a) to synthesize textures; (d) - coupled reaction-diffusion and anisotropic diffusion are used to synthesize enhanced quality textures.

- Variational formulation
- Discretize in the time domain
- Discretize in the spatial domain
- Refine finite elements and basis functions
- Construct and solve the resulting linear system

The finite element solution based on Galerkin discretization, use a variational approach to first generate a weak integral formulation, and then a discretization into a linear system.. For time-varying systems, the time domain is discretized using a semi-implicit backward Euler method. Recursive subdivision techniques are adopted to refine finite elements and basis functions. The mass matrix, the stiffness matrix, and the force vectors are constructed after evaluating each finite element. Finally textures are generated by constructing and solving a linear system.

Various textures can be synthesized by choosing different values of the parameters in the governing equations. In this paper, reaction-diffusion is used to form stable patterns such as spots and stripes when two or more chemicals diffuse at different rates and react with one another, and different coefficients in the PDEs will influence the formation of textures. Anisotropic diffusion is coupled with reaction-diffusion equations to smooth the noise of the input surface mesh, and thereby toncrease the accuracy/quality of texture synthesis.

The remainder of this paper is organized as follows: Section 2 summarizes previous work related to texture synthesis, and finite element simulations; Section 3 explains the algorithm of Galerkin based finite element method solutions; Section 4 discusses the two coupled physical systems, reaction-diffusion and anisotropic diffusion; The final section presents our conclusion and future work.

2 Previous Work

Previous work on texture synthesis with high visual or numerical accuracy focuses on pattern generation and texture mapping, including statistical, non-parametric, as well as optimization-based techniques. In recent years, many texture synthesis techniques have been developed.

Reaction-diffusion: The reaction-diffusion equations were proposed as a model of biological pattern formation for texture synthesis. The traditional reaction-diffusion systems are extended by allowing anisotropic and spatially non-uniform diffusion, as well as multiple competing directions of diffusion [4]. There have been some attempts searching for different ways to generate textures on arbitrary surfaces based on FDM. Reaction-diffusion textures are generated to match the geometry of an arbitrary polyhedral surface by creating a mesh over a given surface and simulating the reaction-diffusion process directly on this mesh [5].

Texture Synthesis: During the past decade, many example-based texture synthesis methods have been proposed, including parametric methods [12, 19, 6, 7], non-parametric methods [13, 14, 20, 27, 28], optimization-based methods [15, 10], and appearance-space texture synthesis [16]. In order to synthesize textures over surfaces based on a given texture example, parametric methods attempt to construct a parametric model of the texture. Differently, non-parametric methods grows the texture one pixel/patch at a time. Optimization-based methods evolve the texture as a whole and improve the quality of the results. Besides texture synthesis on surfaces, various techniques have also been developed for solid texture synthesis [17, 18].

Anisotropic Diffusion: The isotropic diffusion method can remove noise, but blurs features such as edges and corners. In order to preserve features during the process of noise smoothing, anisotropic diffusion [22] was proposed by introducing a diffusion tensor. Generally, a Gaussian filter is used to calculate the anisotropic diffusion tensor before smoothing, but it also blurs features. Bilateral filtering [23], a nonlinear filter combining domain and range filtering, was introduced to solve this problem. Anisotropic diffusion can be used for fairing out noise both in surface meshes and functions defined on the surface [1, 24].

Simulation Using FEM: FEM has been used extensively in solving physically based problems. A finite element solver, CHARMS (conforming, hierarchical, adaptive refinement methods), constructs a framework for adaptive simulation by refining basis functions instead of refining elements [2]. An automated procedure [3] to generate a 3D finite element model of an individual patient's mandible with dental implants inserted was presented. Various methods of image processing, geometric modeling and finite element analysis were combined and extended. The deformation field between 3D images was computed by locally minimizing the sum of the squared differences between the images to be matched [11]. Nonlinear FEM using mass lumping was applied to produce a diagonal mass matrix that allows real time computation, and dynamic progressive meshes were generated to provide detailed information while minimizing computation [21].

3 Galerkin Based Finite Element Method

Textures can be generated by simulating physical phenomena, which are simplified into mathematical models represented by PDEs. Although FDM and its variants have been used extensively to solve PDEs, it is still challenging to synthesize textures directly over surfaces. FEM is a more stable and robust method in solving PDEs over arbitrary manifold surfaces.

Given PDEs with the required property parameters and the corresponding boundary and initial conditions, FEM tends to solve the weak form of these governing equations. The trial space and the test space are introduced. After the spatial and temporal discretization, each element is analyzed, and the recursive subdivision is used to refine the elements and basis functions. In the Galerkin method, the same format is adopted to construct the trial function and the test function. The variational formulations are rewritten by plugging the trial and test functions into the weak form, and are modified with the boundary conditions. In the end, a simplified linear system is built by uniting all the finite elements,

$$Kx = b \tag{1}$$

where x is the unknown vector. The matrix K and the vector b are calculated over the surface domain, which is discretized into small elements such as triangles. The trial and test functions are defined in the same format as a linear combination of basis functions, whose weights are elements of the unknown vector x. As a result, textures are generated by solving the linear system.

3.1 Variational Formulation

A generalized PDE over surface $\Omega \subset \mathbb{R}^3$ is shown in Equation (2), which can represent different physical phenomena by choosing corresponding variables and coefficients, such as the reaction-diffusion and the anisotropic diffusion.

$$\frac{\partial u}{\partial t} = C_0 \mathrm{div}(C_1 \nabla u) + C_2 u + C_3 \tag{2}$$

where div and ∇ are the divergence and the gradient operator over surface Ω (see [26] for their definitions), u represents different variables for various physical problems. u can be a scalar, for example, u is the concentration of a chemical in the reaction-diffusion equations. u can also be a vector in $\mathbb{R}^3$ such as the geometric position or function vectors at each vertex on Ω in the anisotropic diffusion equation. C_0, C_1, C_2, and C_3 are coefficients which can be functions of u, or just constants. C_1 could be a scalar or a 3×3 matrix.

Suppose u, ν are selected from the *trail space* and the *test space* respectively, the inner product of u and ν is defined as follows,

$$(u, \nu) = \int_{\Omega} u\nu dx. \tag{3}$$

By using the Divergence Theorem and the Partial Integration Rule, Equation (2) can be written in a variational form as in Equation (4),

$$\begin{aligned}\left(C_0^{-1}\frac{\partial u}{\partial t}, \nu\right) &= -(C_1\nabla u, \nabla\nu) + (C_0^{-1}C_2 u, \nu) \\ &+ (C_0^{-1}C_3, \nu).\end{aligned} \tag{4}$$

The variational form is the starting point for the following spatial and temporal discretizations.

3.2 Discretization

For time-varying systems, the variational form needs to be discretized in the temporal space as well as in the spatial space. A *semi-implicit backward Euler method* is used for temporal discretization. For the spatial discretization, functions over the integration domain can be refined using the recursive subdivision schemes [1].

Spatial Discretization: The variational problem in Equation (4) is discretized in a function space which is defined by the limit of Loop's recursive subdivision. The function is locally parameterized in our finite element space, which is spanned by C^1 smooth quartic box spline basis functions.

Temporal Discretization: For time-varying systems, we have to discretize Equation (4) in the temporal space. Two issues need to be addressed for the temporal discretization: the choice of the time step, and the decision of which term needs to analyzed implicitly and which term needs to be handled explicitly. Here a *semi-implicit backward Euler discretization* is chosen,

$$\begin{aligned}\Big(\frac{u^{n+1} - u^n}{C_0\tau}, \nu\Big) &= -(C_1\nabla u^{n+1}, \nabla\nu) \\ &+ (C_0^{-1}C_2 u^{n+1}, \nu) + (C_0^{-1}C_3, \nu),\end{aligned} \tag{5}$$

where τ is the time step, u at $t = (n+1)\tau$ is derived from u at $t = n\tau$. Equation (5) is rewritten as follows,

$$\begin{aligned}(C_0^{-1}u^{n+1}, \nu) + [(C_1\nabla u^{n+1}, \nabla\nu) - (C_0^{-1}C_2 u^{n+1}, \nu)]\tau \\ = (C_0^{-1}u^n, \nu) + \tau(C_0^{-1}C_3, \nu).\end{aligned} \tag{6}$$

3.3 Element and Basis Refinement

For each vertex of a control mesh, its basis function is defined by the limit of the Loop's subdivision for the zero control values everywhere except at itself where it is one [1]. The recursive subdivision schemes are used to refine elements and basis functions, and smooth surfaces are generated via a limit procedure of an iterative refinement. Here, an approximating algorithm proposed by Loop [8] is adopted. C^2 limit surfaces are generated except for some extraordinary points at where C^1 continuity is achieved.

If all the vertices of a triangle have a valence of 6, then the triangle is called regular, otherwise it is irregular. A regular patch is controlled by twelve basis functions with explicit polynomial representations. For an irregular patch, the mesh needs to be subdivided repeatedly until the parameter values of interest are inside a regular one [9].

The basis functions are defined by the same recursive scheme. For each vertex of a control mesh, we associate it with a basis function, which is defined as the limit of the Loop's subdivision from control value one at the vertex and control value zero at every other vertices. Each triangle patch is defined locally by only a few related basis functions. Triangles can be grouped according to their vertex valences. All triangles with the same vertex valences have the same set of related basis functions, which only need to be calculated once. Therefore, the computation costs in the numerical integration can be reduced.

3.4 Linear System Construction

The Galerkin approximation is applied to our variational formulation. The trial and test functions are defined in the same format - a linear combination of basis functions,

$$u = \sum_{i=1}^{N} u_i \phi_i, \qquad \nu = \sum_{i=1}^{N} \nu_i \phi_i. \tag{7}$$

Substitute Equation (7) into Equation (6), and rewrite it for each ν_j, we obtain

$$(M + \tau L)u^{n+1} = Mu^n + \tau F, \tag{8}$$

where the mass matrix M, the stiffness matrix L, and the force vector F can be calculated as follows,

$$M_{ij} = (C_0^{-1}\phi_i, \phi_j), \tag{9}$$

$$L_{ij} = (C_1 \nabla\phi_i, \nabla\phi_j) - (C_0^{-1}C_2\phi_i, \phi_j), \tag{10}$$

$$F_j = (C_0^{-1}C_3, \phi_j). \tag{11}$$

The resulting linear system arising in each time step τ can be solved by a *preconditioned conjugate gradient method*. After each u_i $(i = 1, ..., N)$ is calculated, we can obtain u by substituting them back into Equation (7).

4 A Coupled System of Reaction-Diffusion and Anisotropic Diffusion

As shown in Figure 2, our finite element solver can be applied to solve multiply finite element problems over arbitrary manifold surfaces when we choose different coefficients C_0, C_1, C_2, C_3 and u in Equation (2). Different systems are simulated using our solver, such as texture synthesis directly on surfaces by solving reaction-diffusion equations. The anisotropic diffusion is coupled to the reaction-diffusion equations in order to generate better quality textures by smoothing the integral domain.

Physical Problems	Equations	Coefficients in the generalized PDEs, Equation (2)			
		(C_0)	(C_1)	(C_2)	(C_3)
Reaction-Diffusion	Eqn.(12)	D_a	1	-sb	$16s$
Reaction-Diffusion	Eqn.(13)	D_b	1	$s(a-1)$	$-s\beta$
Reaction-Diffusion	Eqn.(16)	1	1	$(1+\mathit{\Gamma} cos\omega_f t)uv-d$	$c+\beta$
Reaction-Diffusion	Eqn.(17)	δ	1	$-u^2$	du
Anisotropic Diffusion	Eqn.(18)	$a(x)^\mu$	$a(x)^{1-\mu}D(x)$	0	0

Fig. 2. Coefficients in the generalized PDEs for different physical problems, reaction-diffusion and anisotropic diffusion.

When we construct the linear system for each application system, we do not need to calculate each term since some coefficients are zero. There are a common inner product term (ϕ_i, ϕ_j) in the mass and stiffness matrices if $C_2 = constant$. This term only needs to be calculated once for each element. For our time-varying systems, we do not update each entry at each time step. In order to speed up the whole process, we take the value from the previous time step for some entries instead of recalculating all of them.

4.1 Reaction-Diffusion

As a biologically motivated method of texture synthesis, the reaction-diffusion is a process in which two or more chemicals diffuse and react with each other to form stable patterns. For a two chemical (a and b) reaction-diffusion system [5],

$$\frac{\partial a}{\partial t} = F(a,b) + D_a \mathit{\Delta} a, \tag{12}$$

$$\frac{\partial b}{\partial t} = G(a,b) + D_b \mathit{\Delta} b, \tag{13}$$

where $F(a,b) = s(16-ab)$, $G(a,b) = s(ab-b-\beta)$, a and b are concentrations, D_a and D_b are diffusion constants, β is the simulation random seed, s is a constant, and t is time.

In this paper, FEM is adopted to solve the PDEs. System (12-13) is weakened by using the divergence theorem and the partial integration, then both trial functions (a and b) and the test function (ν) are discretized in the same format. The reformulated variational form can be written as,

$$\begin{aligned}
&(a^{n+1}, \nu) + (s\tau b^n a^{n+1}, \nu) + (D_a \tau \nabla a^{n+1}, \nabla \nu) \\
&\qquad = (a^n, \nu) + (16 s\tau, \nu), \qquad (14)\\
&(b^{n+1}, \nu) - (s\tau a^n b^{n+1}, \nu) + (D_b \tau \nabla b^{n+1}, \nabla \nu) \\
&\qquad + (s\tau b^{n+1}, \nu) = (b^n, \nu) - (s\beta\tau, \nu), \qquad (15)
\end{aligned}$$

where τ is the time step, and a^n is the concentration at time $t = n\tau$. Equation (14-15) can be rewritten into a linear system in Equation (8). The mass matrix, the stiffness matrix and the force vector of Equation (14) are as follows,

$$\begin{aligned}
M_{ij} &= (D_a^{-1}\phi_i, \phi_j), \\
L_{ij} &= (\nabla\phi_i, \nabla\phi_j) - (sbD_a^{-1}\phi_i, \phi_j), \\
F_j &= (16sD_a^{-1}, \phi_j).
\end{aligned}$$

Similarly, the mass matrix, the stiffness matrix and the force vector of Equation (15) are as follows,

$$\begin{aligned}
M_{ij} &= (D_b^{-1}\phi_i, \phi_j), \\
L_{ij} &= (\nabla\phi_i, \nabla\phi_j) + (saD_b^{-1}\phi_i, \phi_j) - (sD_b^{-1}\phi_i, \phi_j), \\
F_j &= -(s\beta D_b^{-1}, \phi_j).
\end{aligned}$$

Given the initial values $a = b = 4.0$, the concentrations of chemical a and b can be obtained at each time step iteratively. Specifically, for each temporal step, the system consisting of equations (14)-(15) is solved iteratively, until a and b achieve their stable states. Figure 3 and 4 show different patterns generated over a sphere surface. Spot and stripe patterns can be controlled by selecting different values of the parameters.

Additionally as shown in Figure 5, we used the same variational algorithm to solve another reaction-diffusion equations,

$$\frac{\partial u}{\partial t} = c + \beta - du + (1 + \Gamma cos\omega_f t)u^2 v + \Delta u, \qquad (16)$$

$$\frac{\partial v}{\partial t} = du - u^2 v + \delta\Delta v. \qquad (17)$$

Figure 6 shows more textures generated by choosing different parameters in the physical phenomena of reaction-diffusion, Equation (12-13) or (16-17).

4.2 Anisotropic Diffusion

If $F(a, b) = 0$, then Equations (12) turns into an isotropic diffusion equation. Another kind of diffusion problem is anisotropic diffusion, which usually couples

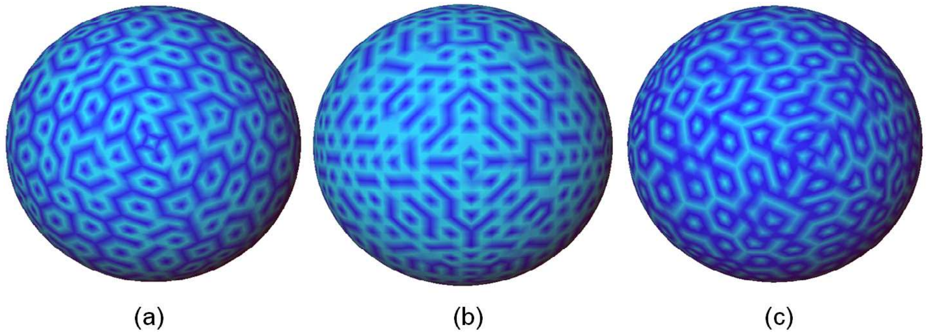

Fig. 3. Textures generated over sphere surfaces using Equation (12-13), where $F(a, b) = s(16 - ab)$, and $G(a, b) = s(ab - b - \beta)$. β = 12.0+/-0.1, D_a = 0.125, D_b = 0.03125. s = 0.025 in (a), s = 0.0375 in (b), and s = 0.05 in (c).

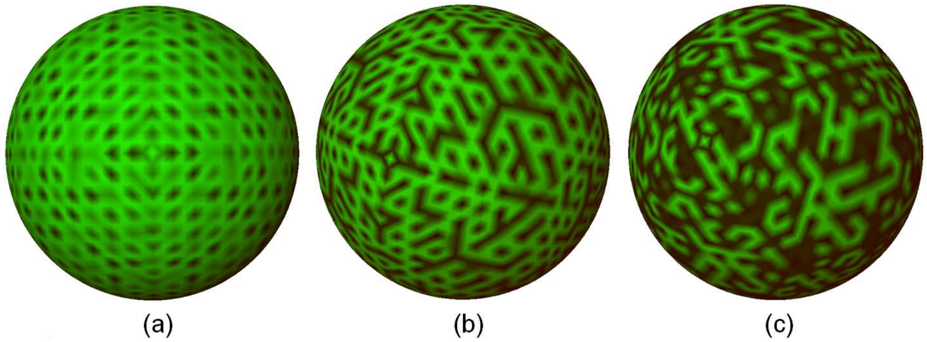

Fig. 4. Textures generated over sphere surfaces using Equation (12-13), where $F(a, b) = a - a^3 - b$, and $G(a, b) = s(a - s_1 b - s_0)$. β = 12.0+/-0.1, D_a = 0.125, D_b = 0.03125. s = 0.025 in (a), s = 0.0375 in (b), and s = 0.05 in (c).

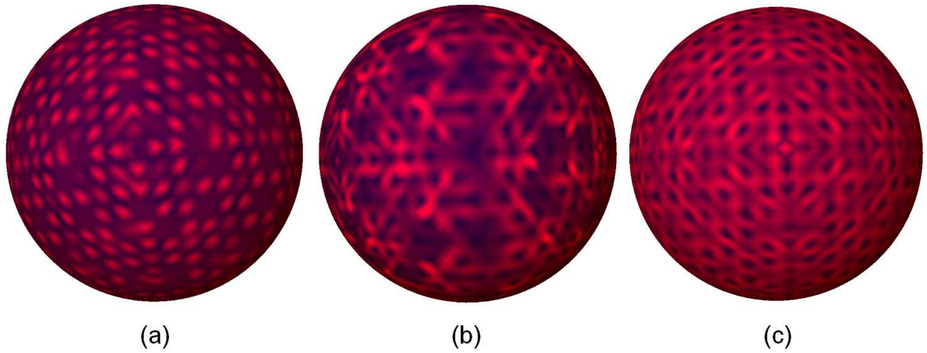

Fig. 5. Textures generated over sphere surfaces using Equations (16-17). β = +/-0.1, $c = 0.5$, $d = 1.5$, $\omega_f = 1.69$. $\delta = 0.0$ in (a), $\delta = 1.0$ in (b), and $\delta = 5.0$ in (c).

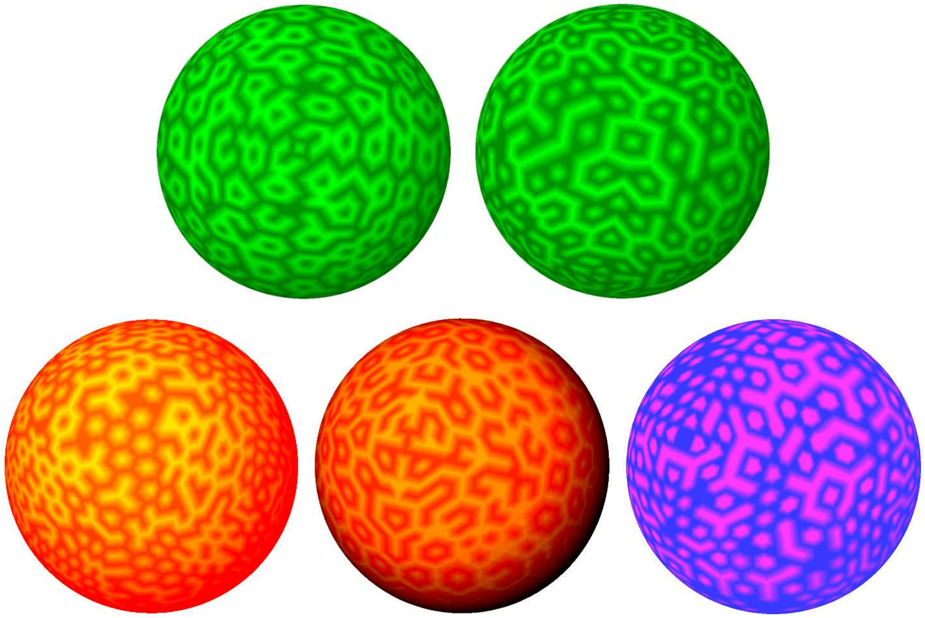

Fig. 6. More textures are generated over sphere surfaces by simulating the physical phenomia of reaction-diffusion.

with the above reaction-diffusion system to generate better textures by smoothing noticeable size variances existing in the surface meshes. In order to remove those artifacts, an anisotropic diffusion tensor is introduced in the governing equation, and the Loop's subdivision techniques are combined with the diffusion model,

$$\frac{\partial x(t)}{\partial t} - a(x)^{\mu} \ \mathrm{div}[(a(x)^{1-\mu} D(x) \nabla x(t))] = 0, \tag{18}$$

where x is the geometric position vector of a point on the surface, t is time, $D(x)$ is a diffusion tensor to enhance sharp features, $a(x)$ is a smooth function which is adaptive to the mesh density, and $\mu \in [0,1]$ is a parameter which changes the smoothing behavior of the equation. $a(x)$ and $D(x)$ are defined in [1]. Let k_1, k_2 be the two principal curvatures, and $e_1(x)$, $e_2(x)$ be the principal directions of the surface at point $x(t)$. Then the diffusion tensor is defined as

$$Dz = \alpha g(k_1) e_1(x) + \beta g(k) 2) e_2(x) + N(x), \tag{19}$$

where $N(x)$ is the normal component of z, and $g(s)$ is

$$g(s) = \begin{cases} 1, & |s| \le \lambda, \\ (1 + \frac{(s-\lambda)^2}{\lambda^2})^{-1}, & |s| > \lambda, \end{cases} \tag{20}$$

λ is a given parameter which detects sharp features. After the spatial and time discretization of the weak form, Equation (18) can be rewritten as follows,

$$\begin{aligned}(a(x^n)^{-\mu}x^{n+1},\nu) + \tau(a(x^n)^{1-\mu}D(x^n)\nabla x^{n+1},\nabla\nu)\\ = (a(x^n)^{-\mu}x^n,\nu),\end{aligned} \tag{21}$$

where ν is the test function, τ is the time step, and n is the step number. The mass matrix, the stiffness matrix and the force vector are as follow,

$$\begin{aligned}M_{ij} &= (a(x)^{-\mu}\phi_i,\phi_j),\\ L_{ij} &= (a(x)^{1-\mu}D(x)\nabla\phi_i,\nabla\phi_j),\\ F_j &= 0.\end{aligned}$$

Our finite element solver can smooth both geometric positions and function values of each vertex on the surface. Anisotropic diffusion is used to improve the quality of the synthesized textures by denoising the surface meshes while preserving geometric features. Figure 1(a-b) show one example.

4.3 A Coupled System

Sometimes noise exists on the geometric surface, therefore in order to generate good quality textures on it, we need to remove the noise. Anisotropic diffusion is a good method for surface fairing because it enhances curve features on the surface by the careful choice of an anisotropic diffusion tensor. In our system, we couple the anisotropic diffusion with the reaction-diffusion together. In other words, the surface is smoothed at the same time when the texture is synthesized.

Figure 7 shows two examples with sharp features, the fandisk model and the mechanical part. The input mesh has some noises on the surface. After applying our coupled system, the surface is smoothed with sharp features preserved and good quality textures are generated as well. Figure 8, 9, and 10 show more results. It is obvious that the noise on the input surface is removed with the preservation of geometric features, and good quality textures are synthesized over the skull, venus and bunny surfaces.

5 Conclusion and Future Work

We have described a stable and robust finite element solver over arbitrary manifold surfaces for a generalized PDE in the format of equation (2). This is used to simulate reaction-diffusion equations coupled with anisotropic diffusion for the synthesis of stable and continuous surface textures without distortions. Different control coefficients of the reaction-diffusion equations are used for the formation of different textures. Additionally, the anisotropic diffusion helps to generate better textures on surfaces by reducing noise in input surface meshes while preserving the surface's geometric features.

There are several directions for future work. Our finite element solvers also work for 3D domain, therefore in the near future we would like to solve the

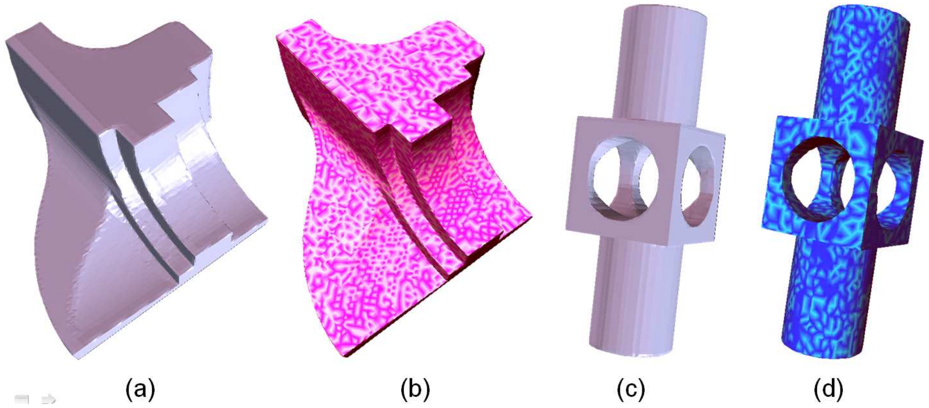

Fig. 7. Textures generated on isosurfaces with sharp feature preservation. (a) - the input surface model of the fandisk; (b) - the texture generated on (a); (c) - the input surface of the mechanical part; (d) - the texture generated on (c).

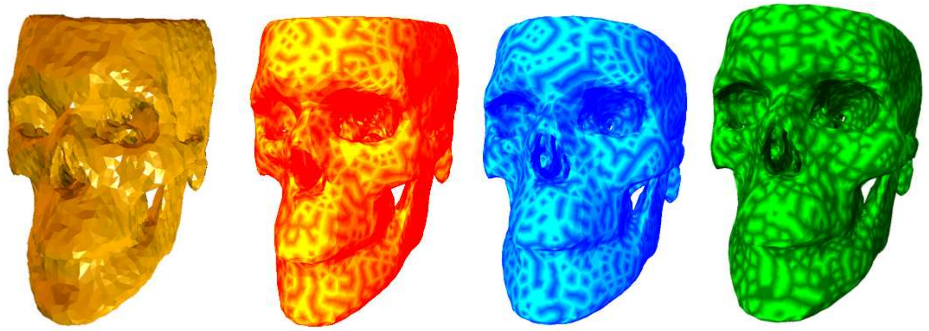

Fig. 8. Textures generated over the skull model by solving the reaction-diffusion governing equations, and the anisotropic diffusion equation is used to smooth noise in the surface mesh (the leftmost picture).

coupled texture synthesis system over a volumetric domain to generate 3D textures directly. Another problem is to tradeoff efficiency vs complexity of pattern generation. In other words, we would like to study if we can solve the PDEs approximately by not updating the matrix for a large number of time steps. Similarly we may not need a semi-implicit time discretization, but perhaps an approximate explicit time discretization to see if we can generate a wide range of patterns or not.

Acknowledgments: C. Bajaj was supported in part by NSF grants EIA-0325550, CNS-0540033, and NIH grants P20-RR020647, R01-GM074258, R01-GM073087 and R01-EB04873. G. Xu was supported in part by National Science Foundation of China grant (60773165), and National Key Basic Research Project of China (2004CB318000).

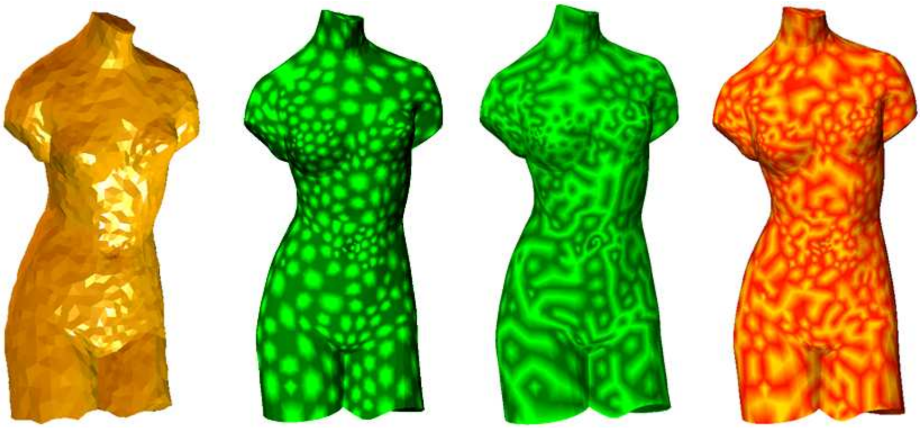

Fig. 9. Textures generated over the venus model by solving the reaction-diffusion governing equations, and the anisotropic diffusion equation is used to smooth noise in the surface mesh (the leftmost picture).

References

1. Bajaj C., Xu G.: Anisotropic Diffusion of Subdivision Surfaces and Functions on Surfaces. ACM Transactions on Graphics. **22(1)** (2003) 4–32
2. Grinspun E., Krysl P., Schroder P.: CHARMS: a simple framework for adaptive simulation. ACM SIGGRAPH. (2002) 281–290
3. Futterling S., Klein R., Strasser W., Weber H.: Automated Finite Element of Human Mandible with Dental Implants. The Sixth International Conference in Central Europe on Computer Graphics and Visualization. (1998)
4. Witkin A., Kass M.: Reaction-Diffusion Texture. ACM SIGGRAPH. (1991) 299–308
5. Turk G.: Generating Textures on Arbitrary Surfaces Using Reaction-Diffusion. ACM SIGGRAPH. (1991) 289–298
6. Wei L., Levoy M.: Texture Synthesis Over Arbitrary Manifold Surfaces. ACM SIGGRAPH. (2001) 355–360
7. Turk G.: Texture Synthesis on Surfaces. ACM SIGGRAPH. (2001) 347-354
8. Loop C.: Smooth Subdivision Surfaces Based on Triangles. Master thesis. Technical Report, Department of Mathematics, University of Utah. (1978)
9. Stam J.: Evaluation Of Loop Subdivision Surfaces. ACM SIGGRAPH. (1998)
10. Balmelli L., Taubin G., Bernardini F.: Space-Optimized Texture Maps. Proceedings of Eurographics. **21(3)** (2002)
11. Ferrant M., Warfield S., Guttmann C., Mulkern R., Jolesz F., Kikinis R.: 3D Image Matching Using a Finite Element Based Elastic Deformation Model. MICCAI 99: Second International Conference on Medical Image Computing and Computer-Assisted Intervention, Germany. (1999) 202–209
12. Heeger D.J., Begen J.R.: Pyramid-based Texture Analysis/Synthesis. ACM SIGGRAPH. (1995) 229–238
13. De Bonet J.S.: Multiresolution Sampling Procedure for Analysis and Synthesis of Texture Images. ACM SIGGRAPH. (1997) 361–368

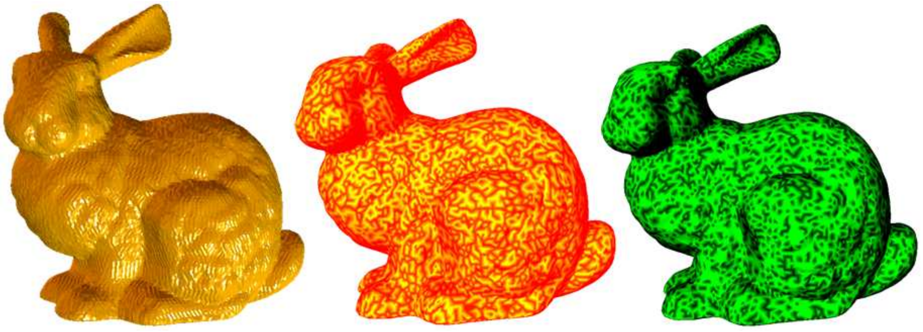

Fig. 10. Textures generated over the rabbit model by solving the reaction-diffusion governing equations, and the anisotropic diffusion equation is used to smooth noise in the surface mesh (the leftmost picture).

14. Wei L.-Y., Levy B.: Fast Texture Synthesis using Three-Structured Vector Quantization. ACM SIGGRAPH. (2000) 479–488
15. Kwatra V., Essa I., Bobick A, Kwatra N.: Texture Optimization for Example-based Synthesis. ACM SIGGRAPH. (2005) 795–802
16. Lefebvre S., Hoppe H.: Appearance-space Texture Synthesis. ACM SIGGRAPH. (2006) 541–548
17. Qin X., Yang Y.-H.: Aura 3D Textures. IEEE Transactions on Visualization and Computer Graphics. **13(2)** (2007) 379–389
18. Kopf J., Fu C.-W., Cohen-Or D., Deussen O., Lischinski D., Wong T.-T.: Solid Texture Synthesis from 2D Exemplars. ACM SIGGRAPH. **26(3)** (2007)
19. Levy B., Mallet J-L.: Non-distorted Texture Mapping for Sheared Triangluated Meshes. ACM SIGGRAPH. (1998) 343–352
20. Soler C., Cani M., Angelidis A.: Hierarchical Pattern Mapping. ACM SIGGRAPH. (2002) 673–680
21. Wu X., Downes M., Goktekin T., Tendick F.: Adaptive Nonlinear Finite Elements for Deformable Body Simulation Using Dynamic Progressive Meshes. Proceedings of Eurographics. **20(3)** (2001)
22. Weickert J.: Anisotropic Diffusion in Image Processing. B. G. Teubner Stuttagart. (1998)
23. Tomasi C., Manduchi R.: Bilateral Filtering for Gray and Color Images. IEEE Computer Vision. (1998) 839–846
24. Clarenz U., Diewald U., Rumpf M.: Nonlinear anisotropic diffusion in surface processing. IEEE Visualization Conference. (2000) 397–405
25. Lin A. L., Hagberg A., Ardelea A., Bertram M., Swinney H. L., Meron E.: Four-Phase patterns in forced oscillatory systems. Phys. Rev. E. **62** (2000) 3790–3798
26. Xu G. and Zhang Q.. Construction of Geometric Partial Differential Equations in Computational Geometry. *Mathematica Numerica Sinica*, 28(4):337–356, 2006.
27. Tong X., Zhang J., Liu L., Wang X., Guo B., Shum H.-Y.: Synthesis of Bidirectional Texture Functions on Arbitrary Surfaces. ACM SIGGRAPH. (2002) 665-672
28. Zhang J., Zhou K., Velho L., Guo B., Shum H.-Y.: Synthesis of Progressively-Variant Textures on Arbitrary Surfaces. ACM SIGGRAPH. (2003) 295-302