

Deformation of Dynamic Surfaces

L.H. You and Jian J. Zhang

National Centre for Computer Animation
Bournemouth University
BH12 5BB
United Kingdom
{lyou, jzhang}@bournemouth.ac.uk

Abstract. Based on dynamic fourth order partial differential equations, we present an iterative finite difference algorithm. With C++ language and OpenGL graphics library, we implement the finite difference algorithm into a user interface and develop shape control parameters, density, damping coefficient, boundary tangents and external forces into user handles for dynamic manipulation of surface deformations. Using the developed user interface, we investigate how these user handles influence the deformation of dynamic surfaces.

Keywords: Surface deformation, dynamic fourth order partial differential equations, iterative finite difference algorithm, influences of user handles.

1 Introduction

Deformation of surfaces can be purely geometric or based on some physical laws. Purely geometric deformation tools change surface shapes mainly by tweaking control points of surfaces represented by Bézier, B-spline or NURBS [1].

In order to follow underlying laws of surface deformations, physics based surface deformation methods were introduced. Depending on whether dynamic effects and time variable are considered, these methods can be classified into static and dynamic.

Static deformation methods of surfaces only consider static material properties. Based on the concepts of elasticity, Kang and Kak presented a finite element analysis method at two resolutions: a coarse resolution for calculations of gross deformations, and a finer resolution for a better and smoother delineation of surface layers [2]. Léon and his co-workers presented an approach for parametric deformations of free-form surfaces which uses a bar network to represent the control polyhedron of a surface and changes the surface shape by deforming the bar network [3, 4].

Dynamic deformation approaches consider the effects of density and damping on deformations and motion and are time dependent. In general, these approaches are based on the equation of motion and solved by some numerical methods such as the finite element method and finite difference method. Terzopoulos and his co-workers proposed a deformable model derived from the theory of elasticity [5, 6] and further incorporated viscoelasticity, plasticity and fracture [7]. Later on, Metaxas and Terzopoulos gave a technique developed from parametrically defined solid primitives, global geometric deformations and local physics-based deformations and used it to build and animate constrained, nonrigid, unibody or multibody objects [8]. Celniker

and Gossard solved the deformable model based on the minimization of an energy functional subjected to user controlled geometric constraints and loads [9]. Gdkbay and zg described a system for the animation of deformable models [10]. Deriving the control equations of motion from Lagrangian mechanics, Guan et al. introduced a deformable curve and surface modeling module and applied it in construction of N-sided patches, surface smooth joining, curve and surface fairing and so on [11]. Terzopoulos and Qin described an interactive modeling system called dynamic NURBS which incorporates mass distributions, internal deformation energies and other physical quantities into the popular NURBS geometric substrate and demonstrated the flexibility of their models in a variety of applications [12]. Qin and Terzopoulos also presented a dynamic NURBS swung surface model which can be formulated constructively from two NURBS profile curves or by applying a nonlinear constraint to a dynamic NURBS surface [13].

Partial differential equation (PDE) based modeling was firstly proposed by Bloor and Wilson [14] and applied in surface generation[15], surface blending[16], shape parameterization of a two-stroke engine [17] and fluid membranes and vesicles[18], and aerodynamic design of a flying wing [19] etc. In recent years, this modeling method has become more and more active. For example, Ugail and Wilson discussed shape parameterization for automatic design optimization based a PDE formulation [20]. Ugail also used the spine of a PDE surface to parameterize the shape of a complicated object [21]. You and Zhang proposed some new resolution methods and applied them in surface blending [22, 23]. Du and Qin combined PDE with the equation of motion and developed a novel modeling approach using popular numerical methods [24, 25].

In this paper, we will present an iterative finite difference algorithm to dynamic fourth order partial differential equations, implement it into a user interface, and investigate how different user handles affect dynamic deformations of surfaces.

2 Dynamic PDEs

Considering the influences of density and damping on deformations and motion, dynamic surfaces can be described with the solution to the following dynamic fourth order partial differential equations together with the corresponding boundary conditions

$$\left\{ b_i \frac{\partial^4}{\partial u^4} + c_i \frac{\partial^4}{\partial u^2 \partial v^2} + d_i \frac{\partial^4}{\partial v^4} + \rho \frac{\partial^2}{\partial t^2} + \eta \frac{\partial}{\partial t} \right\} x_i(u, v, t) = f_i(u, v, t) \quad (1)$$

$(i = 1, 2, 3)$

where u and v are parametric variables, t is a time variable, x_1 , x_2 and x_3 are position functions, b_i , c_i and d_i are shape control parameters, ρ is the density, η is the damping coefficient, $f_1(u, v, t)$, $f_2(u, v, t)$ and $f_3(u, v, t)$ are force components acting in the directions x_1 , x_2 and x_3 , respectively.

Surface patches can be defined by four boundary curves and the boundary tangents of the surfaces at these boundary curves. Therefore, the boundary conditions used for solving the above dynamic partial difference equations are

$$\begin{aligned}
 u=0 \quad x_i &= b_{i0}(v, t) & \frac{\partial x_i}{\partial u} &= b_{i1}(v, t) \\
 u=1 \quad x_i &= b_{i2}(v, t) & \frac{\partial x_i}{\partial u} &= b_{i3}(v, t) \\
 v=0 \quad x_i &= b_{i4}(u, t) & \frac{\partial x_i}{\partial v} &= b_{i5}(u, t) \\
 v=1 \quad x_i &= b_{i6}(u, t) & \frac{\partial x_i}{\partial v} &= b_{i7}(u, t)
 \end{aligned} \tag{2}$$

($i = 1, 2, 3$)

where $b_{i0}(v, t)$, $b_{i2}(v, t)$, $b_{i4}(u, t)$ and $b_{i6}(u, t)$ are boundary curves, and $b_{i1}(v, t)$, $b_{i3}(v, t)$, $b_{i5}(u, t)$ and $b_{i7}(u, t)$ are boundary tangents.

3 Iterative Finite Difference Algorithm

Partial differential equations (1) subjected to boundary conditions (2) are difficult to solve. For most cases, closed form analytical solutions do not exist. In order to address this problem, in this section, we introduce an iterative finite difference algorithm. Using the mesh indicated in Fig. 1 and the central difference approximation, the difference equations of different orders of partial derivatives of the position functions x_i with respect to the parametric variables u and v in Eqs. (1) and (2) can be written as

$$\begin{aligned}
 \left(\frac{\partial x_i}{\partial u}\right)_0^{t+\Delta t} &= \frac{x_{i1}^{t+\Delta t} - x_{i3}^{t+\Delta t}}{2h} \\
 \left(\frac{\partial x_i}{\partial v}\right)_0^{t+\Delta t} &= \frac{x_{i2}^{t+\Delta t} - x_{i4}^{t+\Delta t}}{2h} \\
 \left(\frac{\partial^4 x_i}{\partial u^4}\right)_0^{t+\Delta t} &= \frac{1}{h^4} \left[6x_{i0}^{t+\Delta t} - 4(x_{i1}^{t+\Delta t} + x_{i3}^{t+\Delta t}) + x_{i9}^{t+\Delta t} + x_{i11}^{t+\Delta t} \right] \\
 \left(\frac{\partial^4 x_i}{\partial u^2 \partial v^2}\right)_0^{t+\Delta t} &= \frac{1}{h^4} \left[4x_{i0}^{t+\Delta t} - 2(x_{i1}^{t+\Delta t} + x_{i2}^{t+\Delta t} + x_{i3}^{t+\Delta t} + x_{i4}^{t+\Delta t}) \right. \\
 &\quad \left. + x_{i5}^{t+\Delta t} + x_{i6}^{t+\Delta t} + x_{i7}^{t+\Delta t} + x_{i8}^{t+\Delta t} \right] \\
 \left(\frac{\partial^4 x_i}{\partial v^4}\right)_0^{t+\Delta t} &= \frac{1}{h^4} \left[6x_{i0}^{t+\Delta t} - 4(x_{i2}^{t+\Delta t} + x_{i4}^{t+\Delta t}) + x_{i10}^{t+\Delta t} + x_{i12}^{t+\Delta t} \right]
 \end{aligned} \tag{3}$$

($i = 1, 2, 3$)

where Δt is a time increment, h is the interval between two adjacent nodes, the subscript 0 on the left-hand side of the equations represents the typical node 0, the first subscript i on the right-hand side of the equations indicates the i th component of the position functions x_i , and the second subscript stands for the index of nodes.

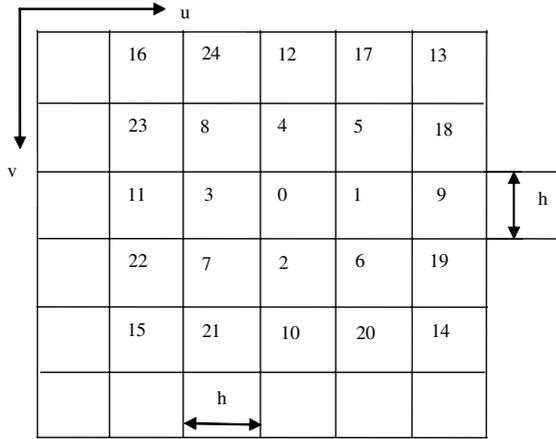


Fig. 1. Typical node of finite difference approximation

Similarly, the first and second partial derivatives of the position functions x_i with respect to the time variable t are transformed into

$$\left(\frac{\partial x_i}{\partial t}\right)_0^t = \frac{x_{i0}^{t+\Delta t} - x_{i0}^{t-\Delta t}}{2\Delta t}$$

$$\left(\frac{\partial^2 x_i}{\partial t^2}\right)_0^t = \frac{x_{i0}^{t+\Delta t} - 2x_{i0}^t + x_{i0}^{t-\Delta t}}{\Delta t^2}$$

$(i = 1, 2, 3)$

Substituting Eqs. (3) and (4) into (1), formulating the finite difference equations for all nodes excluding those on the boundaries, and removing $x_{i0}^{t-\Delta t}$ by means of the relation between the velocity and the displacement

$$V_{i0}^t = \frac{x_{i0}^t - x_{i0}^{t-\Delta t}}{\Delta t} \quad (i = 1, 2, 3)$$

we obtain the following linear algebra equations

$$\begin{aligned} & \left[4(3b_i + 2c_i + 3d_i)\Delta t^2 + (2\rho + \eta\Delta t)h^4\right]x_{i0}^{t+\Delta t} - 4(2b_i + c_i)\Delta t^2(x_{i1}^{t+\Delta t} + x_{i3}^{t+\Delta t}) \\ & - 4(c_i + 2d_i)\Delta t^2(x_{i2}^{t+\Delta t} + x_{i4}^{t+\Delta t}) + 2c_i\Delta t^2(x_{i5}^{t+\Delta t} + x_{i6}^{t+\Delta t} + x_{i7}^{t+\Delta t} + x_{i8}^{t+\Delta t}) + 2b_i\Delta t^2 \\ & (x_{i9}^{t+\Delta t} + x_{i11}^{t+\Delta t}) + 2d_i\Delta t^2(x_{i10}^{t+\Delta t} + x_{i12}^{t+\Delta t}) = 2\Delta t^2 h^4 f_{i0}^t + (2\rho + \eta\Delta t)h^4 x_{i0}^t + (2\rho \\ & - \eta\Delta t)\Delta t h^4 V_{i0}^t \end{aligned}$$

$(i = 1, 2, 3)$

Substituting Eq. (3) into (2), the boundary conditions are changed into the finite difference equations below

$$\begin{aligned}
 u = 0 \quad x_{i0}^{t+\Delta t} &= b_{i0}(v_0, t + \Delta t) & \frac{x_{i1}^{t+\Delta t} - x_{i3}^{t+\Delta t}}{2h} &= b_{i1}(v_0, t + \Delta t) \\
 u = 1 \quad x_{i0}^{t+\Delta t} &= b_{i2}(v_0, t + \Delta t) & \frac{x_{i1}^{t+\Delta t} - x_{i3}^{t+\Delta t}}{2h} &= b_{i3}(v_0, t + \Delta t) \\
 v = 0 \quad x_{i0}^{t+\Delta t} &= b_{i4}(u_0, t + \Delta t) & \frac{x_{i2}^{t+\Delta t} - x_{i4}^{t+\Delta t}}{2h} &= b_{i5}(u_0, t + \Delta t) \\
 v = 1 \quad x_{i0}^{t+\Delta t} &= b_{i6}(u_0, t + \Delta t) & \frac{x_{i2}^{t+\Delta t} - x_{i4}^{t+\Delta t}}{2h} &= b_{i7}(u_0, t + \Delta t)
 \end{aligned} \tag{7}$$

(i = 1, 2, 3)

where the subscript 0 stands for the typical node on the boundaries.

Solving Eqs. (6) and (7) for each time instant, the values of the position functions x_i at all nodes are determined and the deformed surface is defined by the new positions of these nodes. This process is repeated until the whole time period is completed.

4 Influences of User Handles

The above finite difference algorithm has been implemented into a user interface, and shape control parameters, density, damping coefficient, force functions and boundary

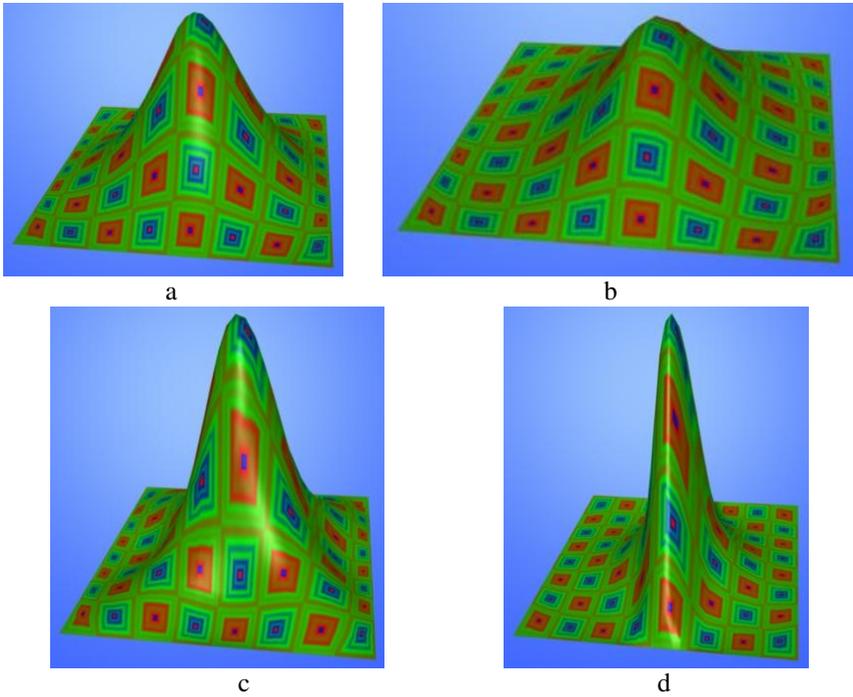


Fig. 2. Influences of shape control parameters

tangents are developed into user handles. In the following, we investigate how these user handles influence shapes of dynamic surfaces.

The basic data of user handles are taken to be: shape control parameters $b_i = d_i = 1$ and $c_i = 2$ ($i = 1, 2, 3$), density $\rho = 0.1$, damping coefficient $\eta = 1$, force functions $f_1 = f_2 = 0$ and $f_3 = 800$, and all the boundary tangents are set to zero. The obtained surface was given in Fig. 2a.

Firstly, we use different shape control parameters and see how the surface shape varies. Setting $b_i = 10$ only leads to the shape in Fig. 2b, changing c_i to -1.5 causes that in Fig. 2c, and taking $d_i = -0.001$ generates the surface in Fig. 2d. It can be concluded that a big value of shape control parameters b_i decreases the deformation, minus values of c_i and d_i raise the size of the deformation, but reduce the deformation region. Between them, the impact of d_i is more obvious.

Next, the effects of boundary tangents on deformations are discussed. Taking different values of boundary tangents, the images in Fig. 3 were generated where Fig. 3a is from $b_{i1}(v, t) = 3$, Fig. 3b is from $b_{i3}(v, t) = 10$, Fig. 3c is from $b_{i1}(v, t) = b_{i3}(v, t) = b_{i5}(u, t) = 3$, and Fig. 3d is from $b_{i1}(v, t) = b_{i5}(u, t) = -3$ and $b_{i3}(v, t) = b_{i7}(u, t) = 3$. Clearly, these different combinations of boundary tangents create quite different appearances of the surface.

Finally, how force functions, density and damping coefficient affect the surface shape were shown in Fig. 4. Keeping all the data same as those in Fig. 2a and only changing the sign of the force, a reverse deformation is produced as depicted in Fig. 4i.

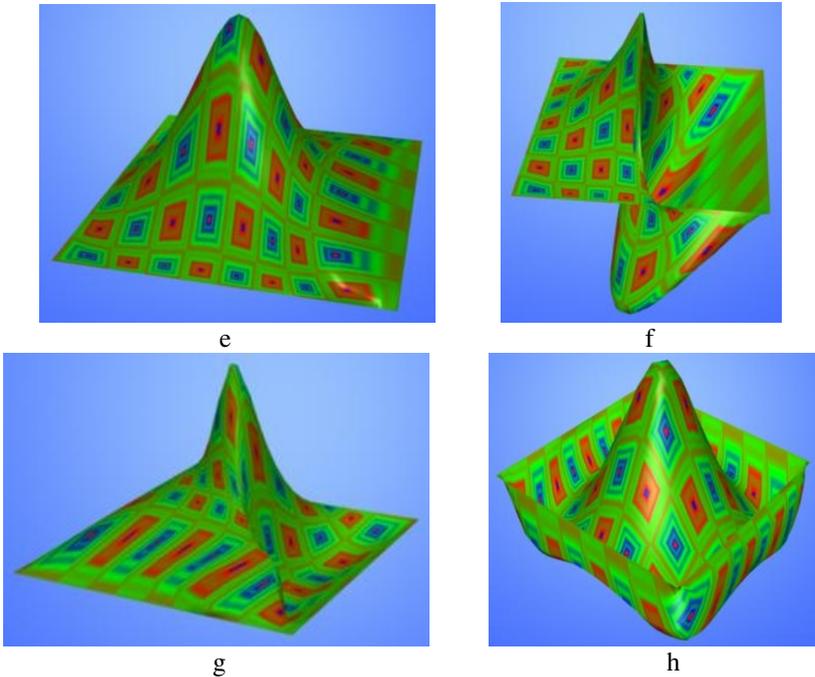


Fig. 3. Influences of boundary tangents

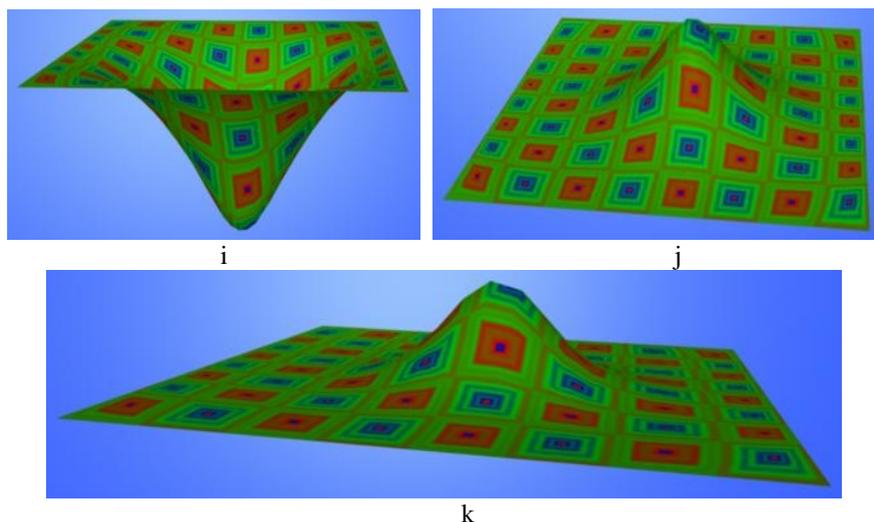


Fig. 4. Influences of force functions, density and damping

Increasing the damping coefficient from 1 to 20, the deformation of the surface becomes more difficult resulting in a much smaller deformation in Fig. 4j. Raising the density of the surface can also decrease the deformation of the surface. Compared the image in Fig. 2a where the density is 0.1, the deformation in Fig. 4k is noticeably reduced due to a rising of the density from 0.1 to 1.

5 Conclusions

In this paper, we have presented an iterative finite difference approximation for dynamic deformations of surfaces which is based on dynamic fourth order partial differential equations. We also implemented the finite difference algorithm into a user interface using C++ and OpenGL graphics library.

With the developed user interface, we examined the influences of user handles on surface deformations. These user handles include shape control parameters, density, damping coefficient, force functions and boundary tangents. They are found to be very effective in shape manipulation of dynamic surface deformations.

Acknowledgements

The authors are very grateful for the donation of the Maya licenses from Alias.

References

1. Farin, G.: Curves and Surfaces for CAGD: A Practical Guide, 5th Edition, Morgan-Kaufmann (2001).
2. Kang, H., Kak, A.: Deforming virtual objects interactively in accordance with an elastic model, Computer-Aided Design, 28(4) (1996) 251-262.

3. Léon, J.C., Veron, P.: Semiglobal deformation and correction of free-form surfaces using a mechanical alternative, *The Visual Computer*, 13 (1997) 109-126.
4. Guillet, S., Léon, J.C.: Parametrically deformed free-form surfaces as part of a variational model, *Computer-Aided Design*, 30(8) 1998 621-630.
5. Terzopoulos, D., Platt, J., Barr, A., Fleischer, K.: Elastically deformable models, *Computer Graphics*, 21(4) (1987) 205-214.
6. Terzopoulos, D., Fleischer, K.: Deformable models, *The Visual Computer*, 4 (1988) 306-331.
7. Terzopoulos, D., Fleischer, K.: Modeling inelastic deformation: viscoelasticity, plasticity, fracture, *Computer Graphics*, 22(4) (1988) 269-278.
8. Metaxas, D., Terzopoulos, D.: Dynamic deformation of solid primitives with constraints, *Computer Graphics (SIGGRAPH '92)*, 26(2) (1992) 309-312.
9. Celniker, G., Gossard, D.: Deformable curve and surface finite-elements for free-form shape design, *Computer Graphics (SIGGRAPH '91)*, 25(4) (1991) 257-266.
10. Güdükbay, U., Özgüç, B.: Animation of deformable models, *Computer-Aided Design*, 26(12) (1994) 868-875.
11. Guan, Z.D., Ling, J., Tao, N., Ping, X., Tang, R.X.: Study and application of physics-based deformable curves and surfaces, *Computers & Graphics*, 21(3) (1997) 305-313.
12. Terzopoulos, D., Qin, H.: Dynamic NURBS with geometric constraints for interactive sculpting, *ACM Transactions on Graphics*, 13(2) (1994) 103-136.
13. Qin, H., Terzopoulos, D.: Dynamic NURBS swung surfaces for physics-based shape design, *Computer-Aided Design*, 27(2) (1995) 111-127.
14. Bloor, M.I.G., Wilson, M.J.: Generating blend surfaces using partial differential equations, *Computer-Aided Design*, 21(3) (1989) 165-171.
15. Bloor, M.I.G., Wilson, M.J.: Complex PDE surface generation for analysis and manufacture, *Computing Suppl.*, (10) (1995) 61-77.
16. Bloor, M.I.G., Wilson, M.J., Mulligan, S.J.: Generating blend surfaces using a perturbation method, *Mathematical and Computer Modelling*, 31(1) (2000) 1-13.
17. Mimis, A.P., Bloor, M.I.G., Wilson, M.J.: shape parameterization and optimization of a two-stroke engine, *Journal of Propulsion and Power*, 17(3) (2001) 492-498.
18. Bloor, M.I.G., Wilson, M.J.: Method for efficient shape parametrisation of fluid membranes and vesicles, *Physical Review E (Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics)*, 61(4) (2000) 4218-429.
19. Sevant, N.E., Bloor, M.I.G., Wilson, M.J.: Aerodynamic design of a flying wing using response surface methodology, *Journal of Aircraft*, 37(4) (2000) 562-569.
20. Ugail, H., Wilson, M. J.: Efficient shape parametrisation for automatic design optimisation using a partial differential equation formulation, *Computers and Structures*, 81 (2003) 2601-2609.
21. Ugail, H.: Spine based shape parameterisation for PDE surfaces, *Computing*, 72 (2004) 195-206.
22. You, L.H., Zhang, J.J., Comninou, P.: Blending surface generation using a fast and accurate analytical solution of a fourth order PDE with three shape control parameters, *The Visual Computer*, 20 (2004) 199-214.
23. You, L.H., Comninou, P., Zhang, J.J.: PDE blending surfaces with C^2 continuity, *Computers & Graphics*, 28 (2004) 895-906.
24. Du, H., Qin, H.: A shape design system using volumetric implicit PDEs, *Computer-Aided Design*, 36 (2004) 1101-1116.
25. Du, H., Qin, H.: Dynamic PDE-based surface design using geometric and physical constraints, *Graphical Models*, 67 (2005) 43-71.