Mar. 2006, Vol.21, No.2, pp.232–237 J. Comput. Sci. & Technol.

Automatic Shape Control of Triangular B-Splines of Arbitrary Topology*

Ying He (贺 英), Xian-Feng Gu (顾险峰), and Hong Qin (秦 洪)

Center for Visual Computing (CVC) and Department of Computer Science, Stony Brook University, Stony Brook NY, 11790-4400, U.S.A.

E-mail: {yhe, gu, qin}@cs.sunysb.edu

Revised December 12, 2005.

Abstract Triangular B-splines are powerful and flexible in modeling a broader class of geometric objects defined over arbitrary, non-rectangular domains. Despite their great potential and advantages in theory, practical techniques and computational tools with triangular B-splines are less-developed. This is mainly because users have to handle a large number of irregularly distributed control points over arbitrary triangulation. In this paper, an automatic and efficient method is proposed to generate visually pleasing, high-quality triangular B-splines of arbitrary topology. The experimental results on several real datasets show that triangular B-splines are powerful and effective in both theory and practice.

Keywords triangular B-splines, arbitrary topology, fairing algorithm

1 Introduction and Motivation

Triangular B-splines, introduced by Dahmen *et al.*[1], are emerging as a novel and powerful tool for shape modeling and interactive graphics, because they can represent, without any degeneracy, complex geometric surfaces defined on open and irregular parametric domains. Using triangular B-splines, or triangular NURBS (the rational generalization of triangular B-splines), users can represent shapes over triangulated planar domains with lower-degree piecewise polynomials (rather than frequently-used tensor-product surface construction over regular domains) that nonetheless maintain higher-order continuity across the boundary of their piecewise patchwork. Prior results have proved that any piecewise polynomial surface over a planar triangulation can be accurately represented in triangular B-splines[1]. Triangular B-splines are even more powerful when being extended and generalized to spherical domain[2,3] and manifold of arbitrary topology[4]. Therefore, triangular B-splines can potentially serve as a geometric standard for product data representation and model conversion in shape design and geometric processing.

Despite their aforementioned geometric advantages and modeling potential over popular tensor-product splines, triangular B-splines have not been widely used in research community and CAD industry. This is mainly because 1) users must deal with a large number of irregularly-distributed control points and their companion knots to make certain non-intuitive decisions on their placements; 2) triangular B-splines have the so-called knot lines, where the surface curvature distribution along the curved triangular boundaries (corresponding to the edges in the domain triangulation) is much worse than other regions. There exist no effective approaches to controlling the overall curvature distribution and improving the shape quality via automatic control-point adjustment.

To overcome these shortcomings of triangular B-splines, this paper develops an automatic algorithm to generate visually pleasing triangular B-splines without the need of any tedious manual operation on control points. Moreover, unlike the existing, classical fairing algorithms, which usually involve the expensive computation of physics-based fair functionals (such as membrane or thin-plate energy), our method solves a simple least square with linear constraints. Therefore, our approach is both fast and robust. Furthermore, our approach works for planar, spherical, and manifold triangular B-splines without any theoretical difficulties. Fig.1 shows an example generated using our automatic shape-fairing algorithm. The input is a C^4 spherical triangular B-spline (shown in Fig.1(b)) with 682 domain triangles (shown in Fig.1(a)). Pay attention to the spline surface marked with red curves which correspond to the edges of spherical triangulation (shown in Fig.1(c)), and the mean curvature plot (shown in Fig.1(d)), the spline surface have high curvature concentrations along the image of edges of the underlying domain triangulation. After automatic fairing, the overall shape only undergoes a small variation (in fact, the shape deviation from the original one is minimized), but the curvature distribution improves significantly (shown in Figs.1(e–g)).

The remainder of this paper is organized as follows. Section 2 reviews the related work on simplex splines and triangular B-splines. Section 3 documents the theoretical background for planar, spherical, and manifold triangular B-splines. Section 4 presents the algorithm to construct smooth triangular B-splines. Section 5 shows our experimental results. Finally, we conclude the paper in Section 6.

*A preliminary version of this paper appeared in Proc. Pacific Graphics 2005, Macau.

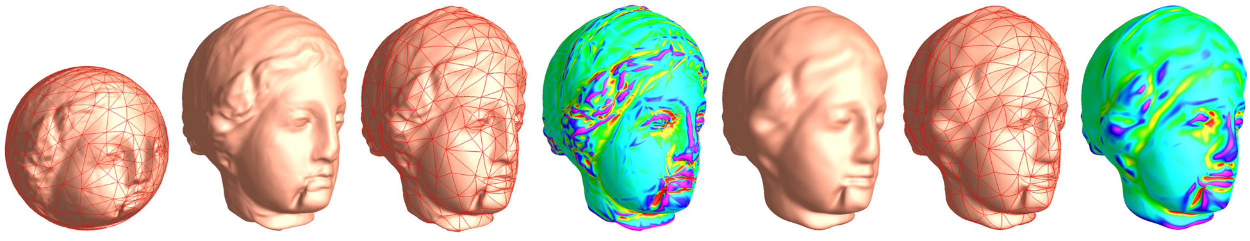


Fig.1. Fairing a spherical triangular B-spline. (a) shows the spherical domain with 682 triangles. (b–d) show a degree 5 (C^4 continuous) spherical spline and its mean curvature plot. Note that the spline surface has high curvature concentration along the image of edges of the spherical triangles. (e–g) show the spline generated by our automatic fairing method. The computational time is 8 seconds on a 3GHz Pentium IV PC. Compared to the surface in (b), the shape of the smooth spline (e) does not change too much, but the curvature distribution improves significantly. The red curves in (c) and (f) correspond to the edges in the spherical triangulation.

2 Previous Work

The theoretical foundation of triangular B-splines lies in the multivariate B-spline, or simplex spline, introduced by de Boor[5] in 1976. Based on the blossom or polar form[6] and B-patch[7], Dahmen *et al.*[1] proposed a general spline scheme in s-dimensional space, which constructs a collection of multivariate B-splines whose linear span comprises all polynomials of degree no more than n. The bivariate case is called triangular B-spline or DMS spline. Due to its elegant construction and many attractive properties for geometric modeling, triangular B-spline has received much attention since its inception. Fong and Seidel[8] presented the first prototype implementation of triangular B-splines and showed several useful properties, such as affine invariance, convex hull, locality, and smoothness. Greiner and Seidel[9] showed the practical feasibility of multivariate B-spline algorithms in graphics and shape design. Pfeifle and Seidel[10] demonstrated the fitting of a triangular B-spline surface to scattered functional data through the use of least squares and optimization techniques. Gormaz and Laurent studied the piecewise polynomial reproduction of triangular B-spline and gave a direct and intuitive proof[11]. Franssen *et al.*[12] proposed an efficient evaluation algorithm, which works for triangular B-spline surfaces of arbitrary degree. He and Qin[13] presented a method of surface reconstruction using triangular B-splines with free knots. Recently, Neamtu[14] described a new paradigm of bivariate simplex splines based on the higher degree Delaunay configurations.

Traditional triangular B-splines are defined on the planar domains. Many researchers have explored the feasible ways to generalize them to be defined on sphere and manifold with arbitrary topology. Alfeld *et al.*[15] presented spherical barycentric coordinates which naturally lead to the theory of Spherical Bernstein-Bézier polynomials (SBB). They showed fitting scattered data on sphere-like surfaces with SBB in [16]. Pfeifle and Seidel[2] presented scalar spherical triangular B-spline and demonstrated its applications for approximating spherical scattered data. Neamtu[17] constructed a functional space of homogeneous simplex splines and showed that restricting the homogeneous splines to a sphere gives rise to the space of spherical simplex splines. He *et al.*[3] presented the rational spherical spline for genus zero shape modeling.

Recently, Gu *et al.*[4] developed a general theoretical framework of manifold splines in which the existing spline schemes defined over planar domains can be systematically generalized to any manifold domain of arbitrary topology (with or without boundaries) using affine structures. They demonstrated the idea of manifold spline using triangular B-splines because of the attractive properties of triangular B-splines, such as arbitrary triangulation, parametric affine invariance, and piecewise polynomial reproduction.

All the existing literatures of triangular B-splines focus on either theoretical foundation or evaluation/data fitting algorithms. No previous work has been done in the surface quality analysis of triangular B-splines. This paper aims at providing such tools for automatic shape control and analysis of triangular B-splines.

3 Construction of Triangular B-Splines

The planar triangular B-spline was proposed by Dahmen *et al.* in [1]. Pfeifle and Seidel successfully generalized the planar triangular B-splines to the spherical domain[2]. Recently, Gu *et al.*[4] systematically built the theoretic framework of manifold spline, which locally is a traditional planar spline, but globally defined on the manifold. They demonstrated manifold splines using triangular B-spline as building block.

The constructions of planar and spherical triangular B-splines are simple and straightforward. In the interests of space, we only briefly introduce the construction of manifold triangular B-spline: given a triangular mesh of arbitrary topology with or without boundaries, we first compute the global conformal parameterization of the domain manifold[18]. Then, we compute a special atlas covering the manifold, such that the transition functions are affine. Next, we define the sub-knots on the manifold directly. Finally, we define basis functions on the chart and assign control point to each basis function.

Triangular B-splines have many valuable properties which are desirable for geometric modeling. For examples, triangular B-splines are piecewise polynomial defined on the parametric domain of arbitrary triangulation. Therefore, the computations of various differential properties are robust and efficient. The degree n triangular B-spline is of C^{n-1} continuous everywhere if there are no degenerate knots. Furthermore, by intentionally placing knots along the edges of the domain triangulation, we can model sharp features easily. The manifold spline of genus g has no more than $2g-2$ singular points while planar and spherical splines do not. Table 1 summarizes the properties of triangular B-splines for shape modeling.

4 Fairing Algorithm

The fairing is of central importance during the design process of free form surfaces. Conventional methods for local and global fairing usually involve a physics based fairness criterion, e.g., membrane energy and thin-plate energy. Note that these fairness functionals involve the integration of the derivatives of $\boldsymbol{F}$ over the parametric domain. Calculating the exact value of the fairness functional is challenging for triangular B-splines, since there is no restriction on the domain triangulation and the sub-knots are also distributed irregularly. In this paper, we propose a new post-processing fairing method which does not need the computation of the complicated double integral. Instead, it only relies on a set of linear constraints of the control points.

Our method is inspired by the knot-line elimination work of Gormaz[19]. Although triangular B-spline has C^{n-1} continuity if there are no degenerate knots, the spline surfaces may not be smooth as expected. The curvature along the images of the edges in the parametric domain is larger than other regions. Fig.2 shows a degree 4 planar triangular B-spline, which is C^3 continuous everywhere. However, the surface is not visually smooth due to the high curvature concentration along the edges of adjacent spline patches. This phenomenon is called "knot line" of the triangular B-splines.

Given a degree n triangular B-spline surface $\boldsymbol{F}(\boldsymbol{u})$ defined on arbitrary triangulation, consider two domain triangles $\Delta(I) = [\boldsymbol{t}_0^I, \boldsymbol{t}_1^I, \boldsymbol{t}_2^I]$ and $\Delta(J) = [\boldsymbol{t}_0^J, \boldsymbol{t}_1^J, \boldsymbol{t}_2^J]$ such that $\Delta(I)$ and $\Delta(J)$ are adjacent. For example, suppose $\boldsymbol{t}_0^I = \boldsymbol{t}_0^J$ and $\boldsymbol{t}_1^I = \boldsymbol{t}_1^J$ (see Fig.3). The sub-knots satisfy $\boldsymbol{t}_{0,i}^I = \boldsymbol{t}_{0,i}^J$ and $\boldsymbol{t}_{1,i}^I = \boldsymbol{t}_{1,i}^J$, for $i = 1, \ldots, n$. Let $\boldsymbol{F}^I$ be the piecewise polynomial restricted on the triangle $\Delta(I)$, i.e., $\boldsymbol{F}^I(\boldsymbol{u}) = \sum_{|\beta|=n} \boldsymbol{c}_{I,\beta} N(\boldsymbol{u}|V_\beta^I)$. Let f^I be the polar form of $\boldsymbol{F}^I$. Similarly, we can define $\boldsymbol{F}^J$ and f^J, respectively (see [20] for the details of polar form). Then, Gormaz proved[19]:

The spline surface $\boldsymbol{F}(\boldsymbol{u})$ has no discontinuity of its n-th derivative along the lines

$$[\boldsymbol{t}_{0,\beta_0}^I, \boldsymbol{t}_{1,\beta_1}^I], \forall \beta, |\beta| = n, \beta_2 \leqslant r$$
$$\textit{iff}\ \ \boldsymbol{c}_{I,\beta} = f^J(V_\beta^I), \forall \beta, |\beta| = n, \beta_2 \leqslant r \qquad (2)$$

where $0 \leqslant r \leqslant n-1$, *and* $V_\beta^I = \{t_{0,0}^I, \ldots, t_{0,\beta_0-1}^I, \ldots, t_{2,0}^I, \ldots, t_{2,\beta_2-1}^I\}$.

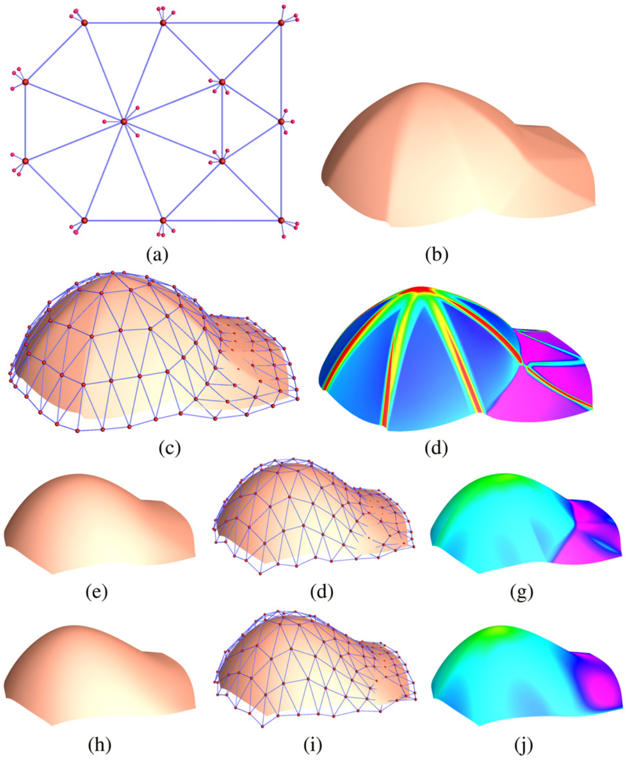


Fig.2. Illustration of our fairing algorithm to a degree 4 planar triangular B-spline. (a) Parametric domain. (b) Spline surface. (c) Control net. (d) Mean curvature plot of the spline surface. (Note that the curvature along the image of edges on the domain triangulation is significantly larger than the vicinity.) (e–g) Fairing the spline surface with $r = 1$. (h–j) Fairing the spline surface with $r = 2$.

(2) defines the affine relations between the control points of $\boldsymbol{F}^I(\boldsymbol{u})$ and $\boldsymbol{F}^J(\boldsymbol{u})$. Given an $r \in [0, n)$, let the control points satisfy (2), then the discontinuity along certain knot lines disappear, and the curvature distri-

Table 1. Properties of Triangular B-Splines

	Triangulation	Local control	Convex hull	Affine invariance	Smoothness	Modeling features	Singular points	Applications
Planar spline	Arbitrary	Yes	Yes	Yes	C^{n-1}	Yes	No	Open surfaces, disk-like topology
Spherical spline	Arbitrary	Yes	No	No	C^{n-1}	Yes	No	Sphere-like, genus zero surfaces
Manifold spline	Arbitrary	Yes	Yes	Yes	C^{n-1}	Yes	Yes	Surfaces of complicated topology

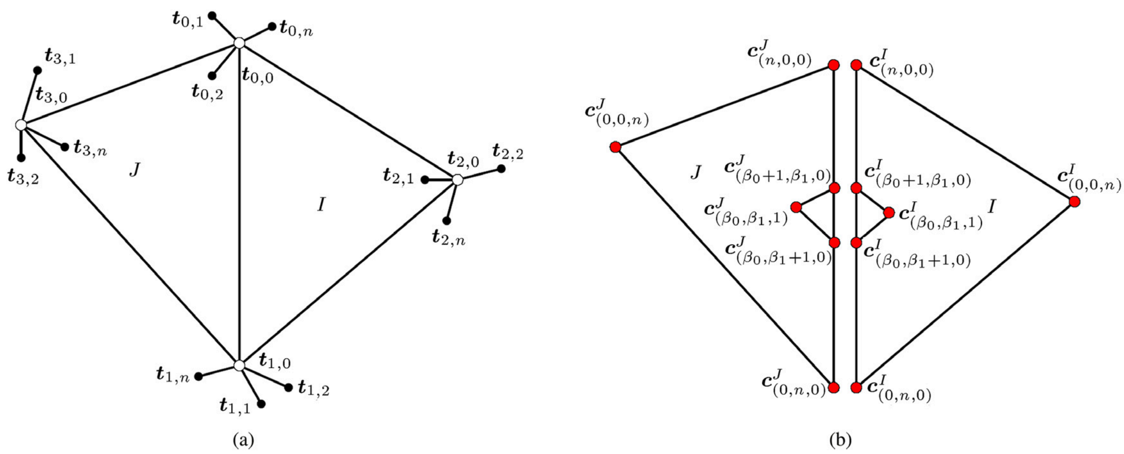


Fig.3. Illustration of (2) for $r = 1$. (a) Parametric domain. (b) Control points.

bution along those lines improves. Fig.3 illustrates the case $r = 1$. For $\beta = (\beta_0, \beta_1, 1)$, (2) is written as

$$\begin{aligned} \boldsymbol{c}^I_{(\beta_0,\beta_1,1)} = {} & \frac{\mathrm{d}(\boldsymbol{t}_{2,0}, \boldsymbol{t}_{1,\beta_1}, \boldsymbol{t}_{3,0})}{\mathrm{d}(\boldsymbol{t}_{0,\beta_0}, \boldsymbol{t}_{1,\beta_1}, \boldsymbol{t}_{3,0})} \boldsymbol{c}^J_{(\beta_0+1,\beta_1,0)} \\ & + \frac{\mathrm{d}(\boldsymbol{t}_{0,\beta}, \boldsymbol{t}_{2,0}, \boldsymbol{t}_{3,0})}{\mathrm{d}(\boldsymbol{t}_{0,\beta_0}, \boldsymbol{t}_{1,\beta_1}, \boldsymbol{t}_{3,0})} \boldsymbol{c}^J_{(\beta_0,\beta_1+1,0)} \\ & + \frac{\mathrm{d}(\boldsymbol{t}_{0,\beta_0}, \boldsymbol{t}_{1,\beta_1}, \boldsymbol{t}_{2,0})}{\mathrm{d}(\boldsymbol{t}_{0,\beta_0}, \boldsymbol{t}_{1,\beta_1}, \boldsymbol{t}_{3,0})} \boldsymbol{c}^J_{(\beta_0,\beta_1,1)} \end{aligned}$$

where $\mathrm{d}(\cdot,\cdot,\cdot)$ is the determinant function. It is easy to verify that (2) is just a linear combination of the control points for $0 \leqslant r \leqslant n-1$.

In the following, we consider the global fairing problem of triangular B-splines. Given a (planar, spherical or manifold) triangular B-spline surface $\boldsymbol{F}(\boldsymbol{u})$, we want to find a faired surface $\widetilde{\boldsymbol{F}}(\boldsymbol{u})$ such that $\widetilde{\boldsymbol{F}}$ approximates the original surface $\boldsymbol{F}$ as much as possible. This leads to the following least square problem:

$$\min_{\tilde{\boldsymbol{c}}} \sum_{I} \sum_{|\beta|=n} \|\tilde{\boldsymbol{c}}_{I,\beta} - \boldsymbol{c}_{I,\beta}\|^2$$

$$\text{subject to } \widetilde{\boldsymbol{c}}_{I,\beta} = f^J(V^I_\beta),\ \forall I, \forall \beta, |\beta| = n, \beta_2 \leqslant r. \quad (3)$$

In the objective function, we minimize the squared distance between the control points of the original and the new spline surface, which implies that the minimal change of the shape. In the constraints, we use an integer $r \leqslant n-1$ to control the fairness of the spline surface. The bigger the value r, the more faired surface we obtain. In our experiments, we can get visually pleasing surfaces with $r = 1$ or $r = 2$.

As mentioned above, the constraints in (3) are just linear equations of the control points. Therefore, (3) is a linear constrained quadratic programming problem which has the following form:

$$\min_{\boldsymbol{x}} \frac{1}{2} \boldsymbol{x}^T \boldsymbol{I} \boldsymbol{x} + \boldsymbol{c}^T \boldsymbol{x} + f \quad \text{subject to } A\boldsymbol{x} = b \quad (4)$$

where $\boldsymbol{I}$ is the identity matrix. In our implementation, we solve the above problem using Lagrange multipliers approach.

5 Experimental Results

We have implemented a prototype system on a 3GHz Pentium IV PC with 1GB RAM. We perform experiments on several models ranging from planar triangular B-splines to manifold triangular B-splines. Table 2 shows the spline configurations and execution times of our test cases.

Table 2. Statistics of Test Cases

Object	Type	n	N_t	N_c	r	Time (s)
Cap	planar	4	13	123	2	< 1
Face	planar	5	251	3181	2	2
Venus	spherical	5	682	8527	2	8
Skull	spherical	5	948	11852	2	16
Dog	spherical	5	656	8202	2	7
Bottle	manifold	3	1889	8513	1	6

Note: n: degree of spline surface; N_t: # of domain triangles; N_c: # of control points; r: smoothness factor. The execution time measures in seconds.

Fig.2 illustrates the fairing algorithm to a planar triangular B-spline. Fig.4 shows example for fairing a spherical triangular B-spline. Compared to the shapes before and after fairing, the curvature concentration phenomena disappear, i.e., the knot-lines are eliminated.

Fig.5 shows examples of smooth triangular B-spline surfaces generated by our fairing algorithm. As shown in Fig.5, we can achieve highly smooth, e.g., C^3 and C^4, triangular B-spline surfaces of various topological types. These results demonstrate that triangular B-splines are both theoretic rigorous and feasible in practice.

6 Conclusion

In this paper, we have proposed an automatic and efficient method to generate visually pleasing, high-quality triangular B-splines of arbitrary topology. Our shape fairing technique works for planar, spherical, and manifold triangular B-splines. Our method is both fast and robust, as only a system of linear equations is solved. Furthermore, the shape deviation is minimized while the overall curvature distribution is significantly improved. Our experimental results on several real datasets have

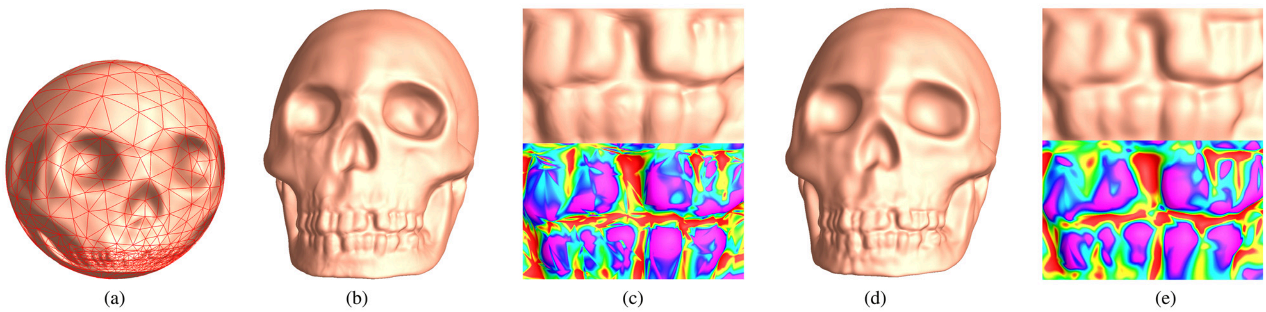


Fig.4. Illustration of our fairing algorithm for a spherical triangular B-spline. (a) Spherical domain. (b) Degree 5 spline with 948 patches. (c) Mean curvature of (b) (red: $H < 0$, cyan: $H > 0$, green: $H \approx 0$). (Pay attention to the high curvature concentration along the image of edges of the spherical triangles.) (d) and (e) After fairing, the curvature distribution improves significantly.

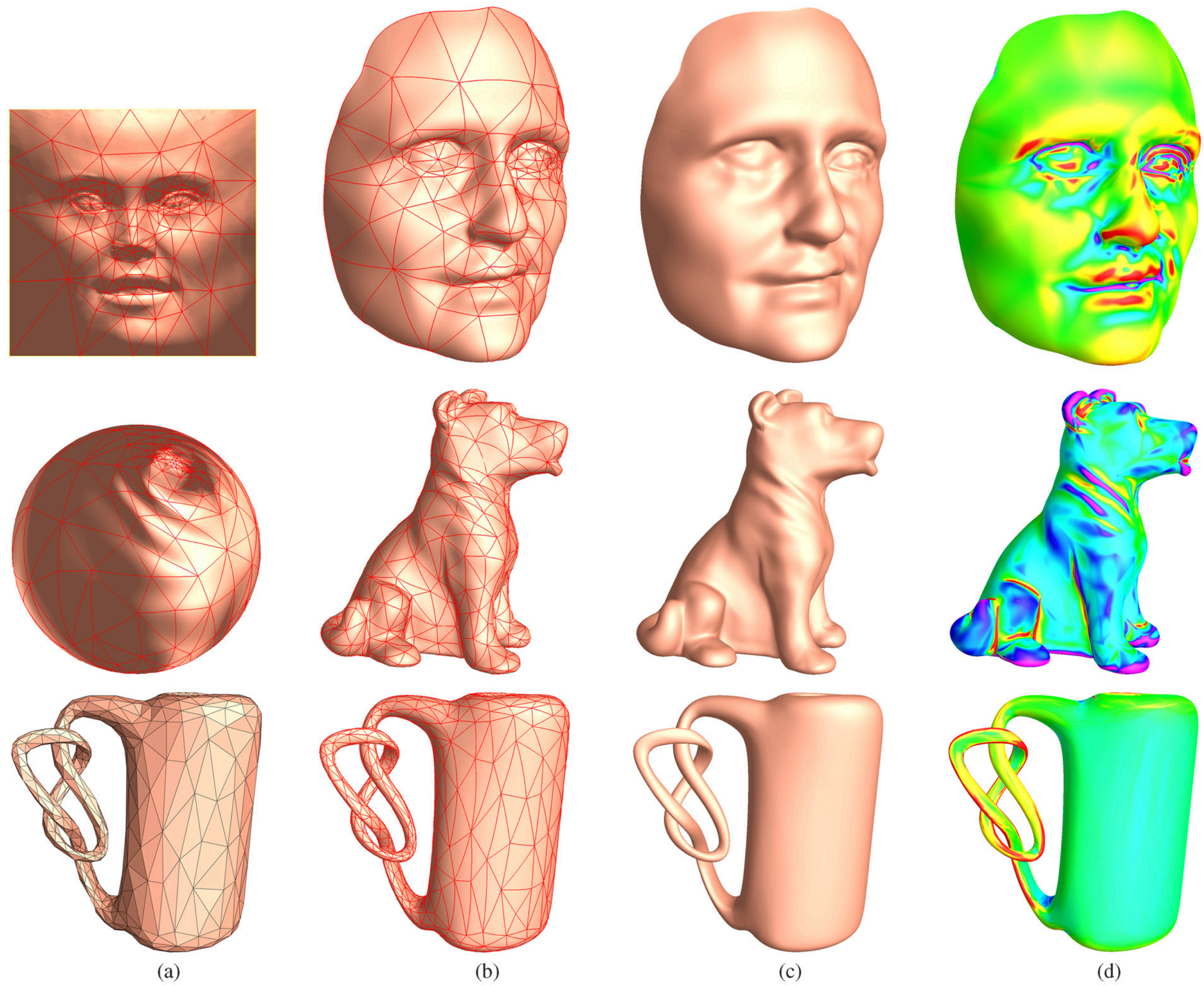


Fig.5. Examples of triangular B-splines. Row 1: C^4 planar spline; Row 2: C^4 spherical spline; Row 3: C^2 manifold spline of genus 2 (the other handle is inside the bottle). (a) shows the parametric domain. The red curves on the spline surfaces (b) correspond to the edges in the domain triangulation (a). (c) and (d) show the spline surfaces and mean curvature plot respectively. Note that there is no restriction on the triangulation of the parametric domain. Those knot-lines (curvature concentration on the image of the edges of domain triangulation) are completely eliminated.

demonstrated that triangular B-splines are powerful and effective in both theory and practice.

Acknowledgement The models are courtesy of Hugues Hoppe, Cindy Grimm, Michael Franssen, Cyberware, Headus, and New York University.

References

[1] Dahmen W, Micchelli C A, Seidel H P. Blossoming begets B-spline bases built better by B-patches. *Mathematics of Computation*, 1992, 59(199): 97–115.

[2] Pfeifle R, Seidel H P. Spherical triangular B-splines with application to data fitting. *Computer Graphics Forum*, 1995, 14(3): 89–96.

[3] He Y, Gu X, Qin H. Rational spherical splines for genus zero shape modeling. In *Proc. International Conference on Shape Modeling and Applications* (*SMI'05*), Boston, USA, 2005, pp.82–91.

[4] Gu X, He Y, Qin H. Manifold splines. In *Proc. ACM Symposium on Solid and Physical Modeling* (*SPM'05*), Boston, USA, 2005, pp.27–38.

[5] C de Boor. Splines as Linear Combinations of B-Splines. A survey, In Approximation Theory II, Academic Press, 1976, pp.1–47.

[6] Ramshaw L. Blossom are polar forms. *Computer Aided Geometric Design*, 1989, 6(4): 323–358.

[7] Seidel H P. Symmetric recursive algorithms for surfaces: B-patches and the de Boor algorithm for polynomials over triangles. *Constr. Approx.*, 1991, 7: 257–279.

[8] Fong P, Seidel H P. An implementation of multivariate B-spline surfaces over arbitrary triangulations. In *Proc. Graphics Interface* (*GI'92*), Vancouver, Canada, 1992, pp.1–10.

[9] Greiner G, Seidel H P. Modeling with triangular B-splines. *IEEE Computer Graphics and Applications*, 1994, 14(2): 56–60.

[10] Pfeifle R, Seidel H P. Fitting triangular B-splines to functional scattered data. In *Proc. Graphics Interface* (*GI'95*), Quebec, Canada, 1995, pp.26–33.

[11] Gormaz R, Laurent P J. Some results on blossoming and multivariate B-splines. Multivariate Approximation: From CAGD to Wavelets, World Sci. Publishing, 1993, pp.147–165.

[12] Franssen M, Veltkamp R C, Wesselink W. Efficient evaluation of triangular B-spline surfaces. *Computer Aided Geometric Design*, 2000, 17(9): 863–877.

[13] He Y, Qin H. Surface reconstruction with triangular B-splines. In *Proc. Geometric Modeling and Processing* (*GMP'04*), Beijing, China, 2004, pp.279–290.

[14] Neamtu M. Bivariate simplex B-splines: A new paradigm. In *Proc. the 17th Spring Conference on Computer Graphics* (*SCCG'01*), Budmerice, Slovakia, 2001, pp.71–78.

[15] Alfeld P, Neamtu M, Schumaker L L. Bernstein-Bézier polynomials on spheres and sphere-like surfaces. Computer Aided Geometric Design, 1996, 13(4): 333–349.

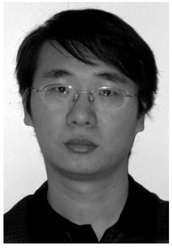
[16] Alfeld P, Neamtu M, Schumaker L L. Fitting scattered data on sphere-like surfaces using spherical splines. *J. Comput. Appl. Math.*, 1996, 73(1-2): 5–43.

[17] Neamtu M. Homogeneous simplex splines. *J. Comput. Appl. Math.*, 1996, 73(1-2): 173–189.

[18] Gu X, Yau S T. Global conformal surface parameterization. In *Proc. the Eurographics/ACM SIGGRAPH Symposium on Geometry Processing* (*SGP'03*), Aachen, Germany, 2003, pp.127–137.

[19] Gormaz R. B-Spline Knot-Line Elimination and Bézier Continuity Conditions. Curves and Surfaces in Geometric Design, MA: A K Peters, Wellesley, 1993, pp.209–216.

[20] Seidel H P. Polar Forms and Triangular B-Spline Surfaces. Euclidean Geometry and Computers, 2nd Edition, World Scientific Publishing Co., 1994, pp.235–286.



Ying He is a Ph.D. candidate in Computer Science Department at Stony Brook University. He obtained his B.S. and M.S. degrees in electrical engineering from Tsinghua University in 1997 and 2000, respectively. His research interests include computer graphics, computer aided geometric design, scientific visualization, computer vision and medical imaging. His current research projects focus on spline-based shape and solid modeling.

Xian-Feng Gu is an assistant professor in computer science at Stony Brook University. He earned his Ph.D. degree in computer science from Harvard University in 2003. He won the NSF CAREER award in 2004. His research interests are computer graphics, computer vision and medical imaging. His major work includes geometry images, global conformal surface parameterization, manifold splines, hyperbolic and projective structures on surface and computational conformal geometry.

Hong Qin is an associate professor of Computer Science Department at Stony Brook University. In 1997, Professor Qin was awarded NSF CAREER Award from the National Science Foundation (NSF). In December, 2000, Professor Qin received Honda Initiation Grant Award. In April, 2001, Professor Qin was selected as an Alfred P. Sloan Research Fellow by the Sloan Foundation. His areas of expertise include geometric modeling, graphics, physics-based simulation, computer aided geometric design, and human-computer interaction. At present, he is an associate editor of IEEE Transactions on Visualization and Computer Graphics (TVCG) and he is also on the editorial board of The Visual Computer (International Journal of Computer Graphics). In 2005, he co-chaired the 23rd Computer Graphics International Conference (CGI 2005).