Computer-Aided Design 81 (2016) 48–60

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Conformal freeform surfaces[☆]

Yi-Jun Yang [a,b], Wei Zeng [c,*], Xiang-Xu Meng [a,b]

[a] School of Computer Science & Technology, Shandong University, Jinan, China
[b] Engineering Research Center of Digital Media Tech., MOE of China, China
[c] School of Computing & Information Sciences, Florida International University, USA

ARTICLE INFO

Article history:
Received 26 September 2015
Accepted 15 September 2016

Keywords:
Hierarchical freeform surfaces
Conformal surfaces
Least square method

ABSTRACT

The conformality of freeform surfaces highly affects their rendering and tessellation results. To improve the conformality of freeform surfaces, a novel freeform surface representation named *hierarchical freeform surfaces* is presented in this paper. The conformality energy of hierarchical freeform surfaces is first formulated and its numerical approximation is then constructed using the composite Simpson's rule. By constructing the parameterization of the initial freeform transformation using the Ricci flow method, the optimal freeform transformation is obtained by the Levenberg–Marquardt method, which is further interleaved with the freeform refinement procedure to generate a hierarchical freeform surface with bounded conformality deviations. Examples are given to show the performance of our algorithm for rendering and tessellation applications.

1. Introduction

Freeform surfaces play an increasingly important role in contemporary graphics and architecture [1]. The results of freeform related applications, such as surface rendering (e.g. texture mapping), surface tessellation, surface sampling, and so on, are highly dependent on the surface parameterization [2–14]. A NURBS surface has an intrinsic rational piecewise polynomial mapping from the 3D surface to the 2D parameter domain (see Fig. 1). By surface reparameterizations [15,12–14], the surface may have infinitely many different parameterizations. Depending on where and how it will be used, one may need to find a suitable or optimal parameterization out of the infinitely many, or to convert the given parameterization into another (more) suitable one [15,12–14]. In many applications, such as texture mapping, surface tessellation, surface matching and registration, it is highly desirable that the parameterization is shape preserving (conformal) i.e. maps an elementary circle of the parameter domain to an elementary circle of the surface. At the same time, the surface modifications (changing the surface control points and/or weights) and surface fitting in reverse engineering both may introduce NURBS surfaces with parameterizations far from being conformal.

Research on conformal mappings has been wide and varied [16–18]. While most of the successes have been reported in generating the conformal mapping of triangle meshes, conformality of complex freeform surfaces has not met with similar achievements. From our point of view, the lack of conformal parameterizations is the bottleneck for freeform surfaces to achieve high quality results for rendering and tessellation applications. Moreover, a conformal parameterization will lead to more robust and stable computations for derivative based algorithms such as surface sampling, surface intersection, curvature computation, and so on [2,7,10,12].

1.1. Related works

We focus on review on freeform surface parameterization methods. For triangle meshes, we concentrate on analyzing the difference between their parameterization and freeform surface parameterization, a detailed review of mesh parameterization techniques being beyond the scope of this article. For the details of mesh parameterization, the reader is recommended to see the survey papers [16,19,20] and the references therein.

The parameterization of triangle meshes has been studied extensively in the last decade and still remains as a hot topic until now [16,21,19,22,23,17,20,18]. The main purpose of the research on the parameterization of triangle meshes is to construct a

[☆] This paper has been recommended for acceptance by Shi-Min Hu.

[*] Correspondence to: 11200 SW 8th Street, Florida International University, School of Computing and Information Sciences, Miami, FL 33199, USA.

E-mail address: wzeng@cs.fiu.edu (W. Zeng).

http://dx.doi.org/10.1016/j.cad.2016.09.003
0010-4485/© 2016 Elsevier Ltd. All rights reserved.

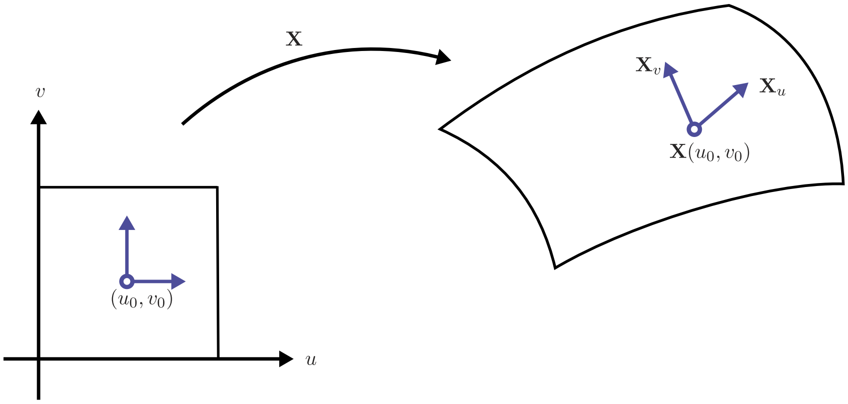


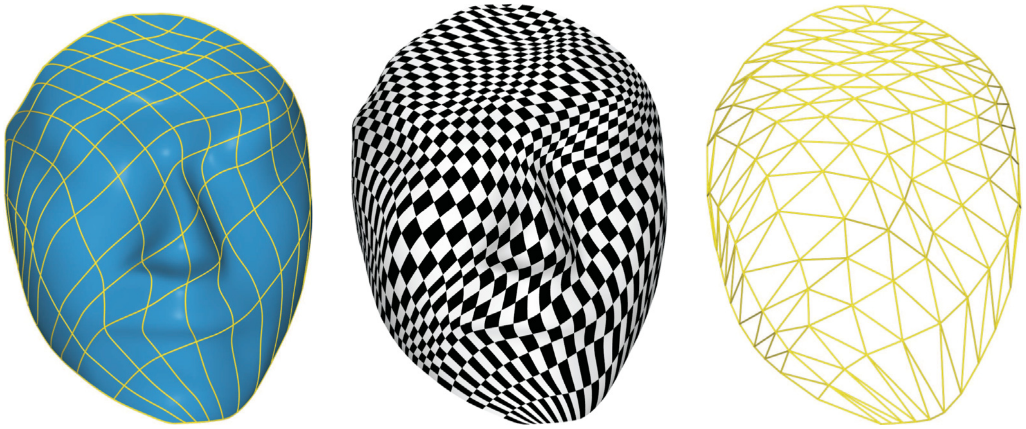
Fig. 1. The intrinsic mapping of a NURBS surface. A 2D point (u_0, v_0) in the parameter domain is mapped to a 3D point $\mathbf{X}(u_0, v_0)$ on the surface. The two partial derivatives $\mathbf{X}_u(u_0, v_0)$ and $\mathbf{X}_v(u_0, v_0)$ at the point $\mathbf{X}(u_0, v_0)$ may be not orthogonal and have different norm lengths.

suitable, bijective mapping between the triangle mesh embedded in 3D and a simple 2D domain, referred to as the parameter space or parameter domain. To minimize the parameterization distortion in terms of either angles or areas, many different algorithms have been proposed in the literature [16,19,23,17,20,18]. As the NURBS surface already has an intrinsic rational polynomial mapping (see Fig. 1) from the 2D parameter domain (a rectangle) to the 3D surface, its parameterization has some specific properties different from the parameterization of triangle meshes. *First*, the research on conformality of freeform surfaces is more complex than that on the triangle meshes. Only a small number of NURBS surfaces are conformal and a general NURBS surface is not. *Second*, the NURBS surface has an intrinsic mapping already and we do not need to construct an initial surface mapping from the 3D surface to the 2D parameter domain any more, which is the case for triangle meshes. *Third*, the parameterization of NURBS surface is a continuous rational polynomial mapping while those of triangle meshes are discrete, usually defined by the correspondence between their vertices and the correspondence of points inside the triangles is obtained from the vertices correspondence by interpolation techniques. If we convert the NURBS surface into a triangle mesh and apply the mesh parameterization method, some parameterization results can be obtained subsequently. However, there is one main drawback for this kind of methods. The resultant surface representation (discrete triangle representation) is not continuous anymore, which is problematic for subsequent algorithms designed for freeform surfaces. Though we can reconstruct the freeform surface by traditional least-square fitting methods from the triangle parameterization, neither the surface shape nor the triangle parameterization are preserved precisely during the fitting procedure, which is not allowed for high accurate freeform related applications.

Recent GPU advances have given rise to the possibility of using freeform surfaces for real-time applications. For example, DirectX 11 adds hardware support for tessellation of arbitrary parametric surfaces. Yet the quality of the tessellation highly depends on its parameterization. In this paper, we give a novel representation of freeform surfaces named *hierarchical freeform surfaces*, which enjoys the hardware support for tessellation as the transformation itself is represented as a 2D freeform surface also. The only difference is to perform the tessellation procedure twice, the first for the transformation surface and the second for the 3D parametric surface.

The aim of this paper is to optimize the conformality of given freeform surfaces for applications such as surface visualization, surface sampling, surface tessellation, surface intersection, curvature computation and so on. Most of all, we want to keep the surface geometry (the surface shape) unchanged and let the resultant surfaces represented as continuous representations, which is preferable and convenient for the algorithms designed for the above mentioned applications. The method presented in this paper utilizes the freeform transformation, which satisfies both the surface shape and representation requirements.

Compared with the quantitative study of the triangle mesh parameterization [24,25,16,26–29,17,30–36,18], little attention has been paid to the freeform surface parameterization until now. The results of rendering and tessellation applications for NURBS surfaces largely depend on the parameterization quality. To perform texture mappings on a NURBS surface, the parametric coordinate of the surface is usually reused as the texture coordinate. If the parameterization is far from being conformal, there will be large distortion of the texture image on the surface (see Fig. 2(b)). To tessellate a NURBS surface, most existing algorithms [5,7,8] map a triangulation of the parameter domain onto the surface. Similar to texture mapping, the final tessellation results largely depend on the surface parameterization (see Fig. 2(c)). A conformal surface parameterization not only preserves the appearance of texture, but also avoids degenerate elements for the tessellation application. Moreover, a conformal parameterization will lead to more robust and stable computations for derivative based algorithms such as surface sampling, surface intersection, curvature computation, and so on [2,7,10,12]. He et al. [27] gave a rational bicubic reparameterization method to improve the parameterization of the approximate Gregory patches such that the new parameterization conforms better to that of the given subdivision surface. Both the explicit representation of the reparameterized surface and the conformality of the final surface are not considered therein. Yang et al. [37] presented an algorithm to improve the Bézier surface parameterization based on Möbius transformations [38,37], which can change only the distribution of iso-parametric curves, but not the shape of them. To obtain more uniform iso-parametric curves, a rational bilinear reparameterization algorithm was also presented in [37]. However, only the uniformity of iso-parametric curves was considered. Furthermore, the rational bilinear reparameterization coefficients are determined by a trivial interpolation method, which is only suitable for a special surface case. To obtain more uniform and orthogonal iso-parameter curves for rational Bézier surfaces, Yang et al. [39] presented an optimization algorithm to minimize the nonlinear energy measuring uniformity and orthogonality deviations using the rational bilinear reparameterizations, which produces a better parameterization with the cost of degree elevation. To further improve the conformality of freeform surfaces, Yang et al. [40] presented another optimization algorithm using the general bilinear transformations. The initial general bilinear transformation is obtained by approximating the conformal mapping of its 3D discretized mesh using a least square method, which is then optimized by the Levenberg–Marquardt



(a) Face surface.

(b) Texture mapping.

(c) Tessellation result.

Fig. 2. Texture mapping and tessellation of the face surface. (a) The face parameterization; (b) the texture mapping of a checkerboard image on the surface; (c) the tessellation of mapping a uniform triangle subdivision of the parameter domain onto the surface.

method. However, both the Möbius transformation and bilinear transformations presented in [37,39,40] do not have sufficient ability to obtain satisfying conformal parameterizations with bounded conformal deviations.

1.2. Algorithm overview

A novel representation of freeform surfaces named *hierarchical freeform surfaces* is presented in this paper. Based on the hierarchical freeform surface, we present an optimization algorithm to improve the conformality of given freeform surfaces. In the optimization procedure, a nonlinear energy measuring the conformality deviations is first formulated and its discrete version is then constructed using the composite Simpson's rule. By constructing the parameterization of the initial freeform transformation using the Ricci flow method, the optimal freeform transformation is obtained by a nonlinear least square method, which is further interleaved with the freeform refinement procedure to generate a hierarchical freeform surface with bounded conformality deviations. Though we utilize NURBS surfaces as the freeform representation in this paper, our method can be generalized to other freeform representations such as T-spline and Gregory surfaces with little modifications. Examples are given to show the performance of our algorithm for tessellation and visualization applications.

The paper is organized as follows. Section 2 describes the hierarchical freeform surfaces and their differential properties. In Section 3, we show how to optimize the conformality of given NURBS surface using the freeform transformations. Several examples are given to show the performance of our algorithm in Section 4 and Section 5 concludes the paper.

2. Hierarchical freeform surfaces

In this paper, we adopt NURBS surfaces as the freeform representation, which can be generalized to other freeform representations with corresponding modifications. For a NURBS surface, its hierarchical freeform representation can be obtained by redefining its parameters (u, v) with a two dimensional transformation $(u, v) = \phi(s, t)$, which is also a freeform surface. The underlying surface geometry is unchanged by defining the hierarchical freeform representation. What changes is the surface parameterization. The basic motivation of introducing hierarchical freeform surfaces is to improve the parameterization of freeform surfaces for the related applications such as surface visualization, surface tessellation, surface sampling, surface blending while keeping their original geometry as well as their parametric continuity. The framework of hierarchical freeform surfaces adapts well to the current GPU architecture by performing the GPU algorithms for traditional freeform surfaces twice. Here we first describe the definition of hierarchical freeform surfaces followed by the analysis of their differential geometry properties.

2.1. Definition

A NURBS surface in three dimensional space is defined by

$$\mathbf{X}(u, v) = \frac{\sum_{i=0}^{n_u}\sum_{j=0}^{n_v} N_i^{p_1}(u) N_j^{q_1}(v)\omega_{i,j}\mathbf{P}_{i,j}}{\sum_{i=0}^{n_u}\sum_{j=0}^{n_v} N_i^{p_1}(u) N_j^{q_1}(v)\omega_{i,j}}, \quad u, v \in [0, 1], \tag{1}$$

where $\mathbf{P}_{i,j}$ are the control points, $\omega_{i,j}$ are the weights, and $N_i^{p_1}(u), N_j^{q_1}(v)$ are the p_1th-degree and q_1th-degree B-spline basis functions defined on the knot vectors

$$\mathbf{U} = \{\underbrace{u_{p_1}, \ldots, u_{p_1}}_{p_1+1}, u_{p_1+1}, \ldots, u_{n_u}, \underbrace{u_{n_u+1}, \ldots, u_{n_u+1}}_{p_1+1}\},$$

and

$$\mathbf{V} = \{\underbrace{v_{q_1}, \ldots, v_{q_1}}_{q_1+1}, v_{q_1+1}, \ldots, v_{n_v}, \underbrace{v_{n_v+1}, \ldots, v_{n_v+1}}_{q_1+1}\},$$

respectively. For the hierarchical freeform representation, (u, v) parameters are subject to the following freeform transformation.

$$(u, v) = \phi(s, t) = \sum_{i=0}^{n_s}\sum_{j=0}^{n_t} N_i^{p_2}(s) N_j^{q_2}(t)\mathbf{Q}_{i,j}, \tag{2}$$

where $\mathbf{Q}_{i,j}$ are the 2D control points, and $N_i^{p_2}(s), N_j^{q_2}(t)$ are the p_2th-degree and q_2th-degree B-spline basis functions defined on their knot vectors

$$\mathbf{S} = \{\underbrace{u_{p_2}, \ldots, u_{p_2}}_{p_2+1}, u_{p_2+1}, \ldots, u_{n_u}, \underbrace{u_{n_u+1}, \ldots, u_{n_u+1}}_{p_2+1}\},$$

and

$$\mathbf{T} = \{\underbrace{v_{q_2}, \ldots, v_{q_2}}_{q_2+1}, v_{q_2+1}, \ldots, v_{n_v}, \underbrace{v_{n_v+1}, \ldots, v_{n_v+1}}_{q_2+1}\},$$

respectively. In this paper, we use a 3 × 3 degree freeform transformation with uniform knot vectors.

2.2. Fundamental Algorithms of hierarchical freeform surfaces

The traditional geometric methods can adapt well with the new representation such as point evaluation, derivative computation, point inversion, surface tangent vector inversion and so on without the explicit representation of the resultant surface. The detailed procedure is described as follows.

1. **Point evaluation**. Given a 2D point (s_0, t_0) in the parameter domain, its corresponding point on surface $\mathbf{X}$ can be obtained by first computing the 2D (u, v) coordinates and substituting them into the original surface parametric equation $\mathbf{X}(\phi(s_0, t_0))$.
2. **Derivative computation**. The derivative of the new representation can be obtained by first computing the derivatives of the NURBS surface and the freeform transformation separately and combining them using the chain rule
$$\mathbf{X}_s = \mathbf{X}_u \cdot u_s + \mathbf{X}_v \cdot v_s \quad \text{and} \quad \mathbf{X}_t = \mathbf{X}_u \cdot u_t + \mathbf{X}_v \cdot v_t.$$
3. **Point inversion**. Given a point $\mathbf{P}$, assumed to lie on the NURBS surface X, point inversion is the problem of finding the corresponding parameter (s_0, t_0), such that $\mathbf{X}(\phi(s_0, t_0)) = \mathbf{P}$. The point inversion of the hierarchical freeform surfaces can be obtained by performing the traditional point inversion method twice, the first for the NURBS surface and the second for the freeform transformation.
4. **Surface tangent vector inversion**. Let (s_0, t_0) be a fixed parameter point and $\mathbf{P} = \mathbf{X}(\phi(s_0, t_0))$ its image on the hierarchical freeform surface. Let $\mathbf{T} = (dx, dy, dz)$ be a vector starting at $\mathbf{P}$ and lying in the tangent plane of the hierarchical freeform surface at $\mathbf{P}$. Denote by $\mathbf{X}_s$ and $\mathbf{X}_t$ the first partial derivatives at (s_0, t_0). If $\mathbf{X}_s \times \mathbf{X}_t \neq 0$, it follows from elementary differential geometry that there exists a vector $\mathbf{W} = (ds, dt)$ in the st plane such that
$$\mathbf{T} = \mathbf{X}_s ds + \mathbf{X}_t dt.$$
Tangent vector inversion is the process of determining the vector $\mathbf{W}$. Surface tangent vector inversion of the hierarchical freeform surfaces can be obtained by first computing the derivatives of the hierarchical freeform surfaces, then obtaining the vector $\mathbf{W} = (ds, dt)$ from the following equations
$$\begin{cases} \mathbf{T} \cdot \mathbf{X}_s = (\mathbf{X}_s \cdot \mathbf{X}_s)ds + (\mathbf{X}_t \cdot \mathbf{X}_s)dt \\ \mathbf{T} \cdot \mathbf{X}_t = (\mathbf{X}_s \cdot \mathbf{X}_t)ds + (\mathbf{X}_t \cdot \mathbf{X}_t)dt. \end{cases}$$

After we give the fundamental algorithms of hierarchical freeform surfaces, their differential properties are described in the following subsection.

2.3. Differential properties

The parameterization in Eq. (1) is characterized by its first fundamental form [41]

$$ds^2 = \mathbf{X}_u \cdot \mathbf{X}_u (du)^2 + 2\mathbf{X}_u \cdot \mathbf{X}_v du dv + \mathbf{X}_v \cdot \mathbf{X}_v (dv)^2,$$

where $\mathbf{X}_u = \frac{\partial \mathbf{X}}{\partial u}$ and $\mathbf{X}_v = \frac{\partial \mathbf{X}}{\partial v}$ are the two partial derivative vectors of the surface $\mathbf{X}$. The first fundamental form describes the metric on the surface $\mathbf{X}$. Let

$$E = \mathbf{X}_u \cdot \mathbf{X}_u, \qquad F = \mathbf{X}_u \cdot \mathbf{X}_v, \qquad G = \mathbf{X}_v \cdot \mathbf{X}_v,$$

and rewrite the coefficients in a symmetric matrix

$$\mathbf{I} = \begin{pmatrix} E & F \\ F & G \end{pmatrix},$$

where E and G give the square length of the two partial derivatives and F measures the orthogonality of the two partial derivatives. Then, we have

$$ds^2 = (du \quad dv)\, \mathbf{I} \begin{pmatrix} du \\ dv \end{pmatrix}.$$

Using the chain rule, we can obtain the partial derivatives of hierarchical freeform surfaces over the new parametric variables (s, t) as follows

$$\mathbf{X}_s^T = \left(\mathbf{X}_u^T \quad \mathbf{X}_v^T\right) \phi_s^T$$
$$\mathbf{X}_t^T = \left(\mathbf{X}_u^T \quad \mathbf{X}_v^T\right) \phi_t^T,$$

where ϕ_s and ϕ_t are the partial derivatives of the 2D transformation surface defined in Eq. (2) over the variables s and t, respectively. The first fundamental terms of the hierarchical freeform surfaces are as follows.

$$\begin{aligned} \widetilde{E} &= \mathbf{X}_s \cdot \mathbf{X}_s = \mathbf{X}_s \mathbf{X}_s^T \\ &= \phi_s \begin{pmatrix} \mathbf{X}_u \\ \mathbf{X}_v \end{pmatrix} \left(\mathbf{X}_u^T \quad \mathbf{X}_v^T\right) \phi_s^T \\ &= \phi_s \begin{pmatrix} E & F \\ F & G \end{pmatrix} \phi_s^T \\ &= \phi_s \mathbf{I} \phi_s^T, \\ \widetilde{F} &= \mathbf{X}_s \cdot \mathbf{X}_t = \mathbf{X}_s \mathbf{X}_t^T \\ &= \phi_s \begin{pmatrix} \mathbf{X}_u \\ \mathbf{X}_v \end{pmatrix} \left(\mathbf{X}_u^T \quad \mathbf{X}_v^T\right) \phi_t^T \\ &= \phi_s \mathbf{I} \phi_t^T, \end{aligned}$$

and

$$\begin{aligned} \widetilde{G} &= \mathbf{X}_t \cdot \mathbf{X}_t = \mathbf{X}_t \mathbf{X}_t^T \\ &= \phi_t \begin{pmatrix} \mathbf{X}_u \\ \mathbf{X}_v \end{pmatrix} \left(\mathbf{X}_u^T \quad \mathbf{X}_v^T\right) \phi_t^T \\ &= \phi_t \mathbf{I} \phi_t^T. \end{aligned}$$

The parameterization of hierarchical freeform surfaces is characterized by its first fundamental form

$$\begin{aligned} d\widetilde{s}^2 &= (ds \quad dt) \begin{pmatrix} \widetilde{E} & \widetilde{F} \\ \widetilde{F} & \widetilde{G} \end{pmatrix} \begin{pmatrix} ds \\ dt \end{pmatrix} \\ &= (ds \quad dt) \begin{pmatrix} \phi_s \\ \phi_t \end{pmatrix} \mathbf{I} \left(\phi_s^T \quad \phi_t^T\right) \begin{pmatrix} ds \\ dt \end{pmatrix}. \end{aligned}$$

Then we obtain the first fundamental matrix of the hierarchical freeform surface as follows

$$\widetilde{\mathbf{I}} = \begin{pmatrix} \phi_s \\ \phi_t \end{pmatrix} \mathbf{I} \left(\phi_s^T \quad \phi_t^T\right).$$

If the hierarchical freeform surface is conformal, we have $\widetilde{F} = 0$ and $\widetilde{E} = \widetilde{G}$ [16], which lead to

$$\begin{cases} \phi_s \mathbf{I} \phi_t^T = 0 \\ \phi_s \mathbf{I} \phi_s^T = \phi_t \mathbf{I} \phi_t^T. \end{cases} \tag{3}$$

The derivatives of transformation ϕ over variables s and t are as follows.

$$\phi_s = (u_s \quad v_s) \quad \text{and} \quad \phi_t = (u_t \quad v_t). \tag{4}$$

Substituting Eqs. (4) into Eqs. (3), we have

$$(u_s \quad v_s) \begin{pmatrix} E & F \\ F & G \end{pmatrix} \begin{pmatrix} u_t \\ v_t \end{pmatrix} = 0, \tag{5}$$

and

$$(u_s \quad v_s) \begin{pmatrix} E & F \\ F & G \end{pmatrix} \begin{pmatrix} u_s \\ v_s \end{pmatrix} = (u_t \quad v_t) \begin{pmatrix} E & F \\ F & G \end{pmatrix} \begin{pmatrix} u_t \\ v_t \end{pmatrix}. \tag{6}$$

From Eq. (5), we have

$$u_t = -\frac{v_t(u_s F + v_s G)}{(u_s E + v_s F)}. \tag{7}$$

Substituting Eq. (7) into Eq. (6), we obtain v_t as follows.

$$v_t = \pm \frac{u_s E + v_s F}{\sqrt{EG - F^2}}. \tag{8}$$

We select the positive sign in Eq. (8) such that the derivatives v_t and u_s have the same variation tendency. By collecting the constraints in Eqs. (7) and (8), we have

$$\mathbf{C}(s,t) = \phi_t^T - \begin{pmatrix} -\frac{F}{\sqrt{EG-F^2}} & -\frac{G}{\sqrt{EG-F^2}} \\ \frac{E}{\sqrt{EG-F^2}} & \frac{F}{\sqrt{EG-F^2}} \end{pmatrix} \phi_s^T = 0, \tag{9}$$

which always holds at a conformal point.

3. Conformal freeform surfaces

Conformal parameterization of NURBS surfaces has several special properties, which are extremely valuable for visualization, tessellation and other derivative-based applications. *First*, conformal parameterization is angle-preserving. Intuitively, suppose γ_1, $\gamma_2 \subset [0,1] \times [0,1]$ are two arbitrary intersecting curves in the parameter domain, with the intersection point as $p = \gamma_1 \cap \gamma_2$. Then they are mapped to intersection curves on the 3D surface $\mathbf{X}$, $\mathbf{X}(p) = \mathbf{X}(\gamma_1) \cap \mathbf{X}(\gamma_2)$. Suppose at the intersection point p, the intersection angle between the tangent vectors $d\gamma_1$, $d\gamma_2$ is θ. $\mathbf{X}$ is conformal, if and only if the intersection angle between the tangent vectors $d\mathbf{X}(\gamma_1)$ and $d\mathbf{X}(\gamma_2)$ is also θ, therefore it is shape preserving. For each (u, v), the tangent vectors to the two iso-parametric curves are orthogonal and have the same norm. In other words, a conformal map is locally isotropic, i.e. maps an elementary circle of the (u, v) domain to an elementary circle of the surface. *Second*, a conformal parameterization will lead to more robust and stable computations for derivative based algorithms such as surface sampling, surface intersection, curvature computation, and so on [2,7].

Given a NURBS surface, this section presents an optimization algorithm to improve its conformality using freeform transformations. To measure the deviation of the current parameterization from conformal parameterizations, the following integral function is adopted

$$J_c(\mathbf{Q}_{i,j}) = \int_0^1 \int_0^1 \left| \phi_t - \begin{pmatrix} -\frac{F}{\sqrt{EG-F^2}} & -\frac{G}{\sqrt{EG-F^2}} \\ \frac{E}{\sqrt{EG-F^2}} & \frac{F}{\sqrt{EG-F^2}} \end{pmatrix} \phi_s \right|^2 dsdt. \tag{10}$$

In general, Eq. (10) is highly nonlinear, and there is no closed-form solution for the freeform transformations. Thus we first discretize the above conformal energy by composite Simpson's rule and then its discretized form can be solved using the nonlinear least square method.

3.1. Numerical method

The terms in Eq. (10) are approximated by a discrete version of the energy using the composite Simpson's rule [42]. To apply the composite Simpson's rule, we uniformly sample the parameter domain by partitioning the u and v intervals into $2l$ and $2k$ subintervals

$$\mathbf{R}_{i,j} = \begin{pmatrix} u_p + \frac{i \cdot (u_{n_u+1} - u_p)}{2l} \\ v_q + \frac{j \cdot (v_{n_v+1} - v_q)}{2k} \end{pmatrix}, \quad i = 0, \ldots, 2l, j = 0, \ldots, 2k. \tag{11}$$

These subdivisions determine the step sizes $h = 1/(2l)$ and $p = 1/(2k)$. Let

$$T = \left| \phi_t - \begin{pmatrix} -\frac{F}{\sqrt{EG-F^2}} & -\frac{G}{\sqrt{EG-F^2}} \\ \frac{E}{\sqrt{EG-F^2}} & \frac{F}{\sqrt{EG-F^2}} \end{pmatrix} \phi_s \right|^2.$$

The resulting approximation has the form

$$J = \widetilde{J} + E_{error},$$

where

$$\begin{cases} \widetilde{J} = \frac{4hp}{9}\{J_1 + 2J_2 + 4J_3 + J_4\} \\ E_{error} = -\frac{h^4 \frac{\partial^4 T(\bar{s},\bar{t})}{\partial s^4} + p^4 \frac{\partial^4 T(\widehat{s},\widehat{t})}{\partial t^4}}{180} \\ J_1 = T(s_{0,0}, t_{0,0}) + 2\sum_{i=1}^{l-1} T(s_{2i,0}, t_{2i,0}) \\ \qquad + 4\sum_{i=1}^{l} T(s_{2i-1,0}, t_{2i-1,0}) + T(s_{2l,0}, t_{2l,0}) \\ J_2 = \sum_{j=1}^{k-1} T(s_{0,2j}, t_{0,2j}) + 2\sum_{j=1}^{k-1}\sum_{i=1}^{l-1} T(s_{2i,2j}, t_{2i,2j}) \\ \qquad + 4\sum_{j=1}^{k-1}\sum_{i=1}^{l} T(s_{2i-1,2j}, t_{2i-1,2j}) + \sum_{j=1}^{k-1} T(s_{2l,2j}, t_{2l,2j}) \\ J_3 = \sum_{j=1}^{k} T(s_{0,2j-1}, t_{0,2j-1}) + 2\sum_{j=1}^{k}\sum_{i=1}^{l-1} T(s_{2i,2j-1}, t_{2i,2j-1}) \\ \qquad + 4\sum_{j=1}^{k}\sum_{i=1}^{l} T(s_{2i-1,2j-1}, t_{2i-1,2j-1}) + \sum_{j=1}^{k} T(s_{2l,2j-1}, t_{2l,2j-1}) \\ J_4 = T(s_{0,2k}, t_{0,2k}) + 2\sum_{i=1}^{l-1} T(s_{2i,2k}, t_{2i,2k}) \\ \qquad + 4\sum_{i=1}^{l} T(s_{2i-1,2k}, t_{2i-1,2k}) + T(s_{2l,2k}, t_{2l,2k}) \end{cases}$$

for some $(\bar{s}, \bar{t})$ and $(\widehat{s}, \widehat{t})$ in the (s, t) parameter domain. We can increase the sampling density l and k to decrease the error term E_{error}. When using $l, k = 10$, the error term has the following form

$$E_{error} = -\frac{\frac{\partial^4 \mathbf{T}(\bar{u},\bar{v})}{\partial u^4} + \frac{\partial^4 \mathbf{T}(\widehat{u},\widehat{v})}{\partial v^4}}{2.88 \times 10^{11}}.$$

For a given NURBS surface, the numerator of E_{error} is bounded and E_{error} is seven order of magnitude smaller than that bound with the sampling density $l, k = 10$. The energy function in Eq. (10) is then discretized as

$$\widetilde{J}_c(\mathbf{Q}_{i,j}) \quad i = 0 \ldots n_s, j = 0 \ldots n_t. \tag{12}$$

Minimizing the Energies in Eq. (12) leads to the following optimization problem

$$\min_{\mathbf{Q}_{i,j}} \widetilde{J}_c, \tag{13}$$

which is a nonlinear least square problem. This target function can be minimized numerically using the Levenberg–Marquardt method, which enjoys robust global convergence property to a local minimum. Each iteration of the Levenberg–Marquardt method needs to evaluate the gradient of function $\widetilde{J}_c$ with respect to the control points $\mathbf{Q}_{i,j}$ of the freeform transformation. For numerical optimization, it is advisable to start from an initial point which is not far away from a local minimum. We determine the initial transformation by the following least square method. *First*, we uniformly sample a grid in the (u, v) parameter domain of the NURBS surface and obtain the 2D conformal mapping $\mathbf{M}$ of their corresponding 3D discretized mesh on the NURBS surface by the Ricci flow method [21]. After the 2D conformal mapping is obtained, the initial freeform transformation can then be constructed by approximating the composite mapping $\mathbf{M} \circ \mathbf{X}$ using the least square method. After the initial freeform transformation is obtained, we optimize the nonlinear least square problem in Eq. (13) using the Levenberg–Marquardt method. The Levenberg–Marquardt algorithm is a popular method for nonlinear least square optimization. It is more robust than the Gauss–Newton algorithm, and is guaranteed to converge to a local minimum [43]. The Levenberg–Marquardt method interpolates between the Gauss–Newton algorithm and the method of gradient descent. The Levenberg–Marquardt method is more robust than the Gauss–Newton algorithm, which means that in many cases it finds a solution even if it starts very far off the final minimum. Levenberg–Marquardt method can also be viewed as Gauss–Newton using a trust region approach. There is no guarantee that it will converge to a global minimum. To find the global minima, we need to use global optimization algorithms, which require a global search approach and are usually time-consuming [44]. Instead, our method selects a good initial point for the Levenberg–Marquardt algorithm, so that it can converge to a local minimum nearby. Such strategy has been successfully applied to other geometric modeling problems involving nonlinear least square optimization [45,46,39,40]. For a specific freeform transformation, the conformality deviation of the resultant hierarchical freeform surface will not be equal to zero even after the nonlinear optimization procedure. To obtain a hierarchical freeform surface with bounded conformality deviations, the nonlinear least square conformality minimization and the freeform transformation refinement interleave during the optimization procedure, which is described in detail in the next subsection.

3.2. Approximation within a specified accuracy

In practice, we want to optimize the transformation of given freeform surfaces such that the hierarchical freeform representation converges to be conformal. The conformality of the hierarchical freeform surface can be checked by uniformly sampling points inside the (s, t) parameter domain and computing the maximal conformal deviation at the sampling points

$$\max_{0\le k\le m} \left| \mathbf{N} \times \frac{\partial \mathbf{X}}{\partial s} - \frac{\partial \mathbf{X}}{\partial t} \right|^2 ,$$

where $\mathbf{N}$ is the unit normal vector of the freeform surface at the current point and $\frac{\partial \mathbf{X}}{\partial s}$ and $\frac{\partial \mathbf{X}}{\partial t}$ are the two partial derivatives of the surface over variables s and t, respectively. If the maximal conformal deviation is larger than the given tolerance ϵ_c, perform the following iterative steps.

1. Refine the freeform transformation.
2. Compute the freeform transformation by the nonlinear least square optimization method.
3. Check the conformal deviation of the new hierarchical freeform surface.
4. If the maximal conformal error is less than the tolerance ϵ_c, exit; otherwise, go to Step 1.

By performing the above refinement and optimization procedures iteratively, we obtain a conformal hierarchical freeform surface with bounded conformal deviations at the end.

4. Experimental results

Our algorithm is implemented on a PC with Intel Duo CPU 3.06 GHz, 2G Memory and Microsoft Visual Studio 2008. To show the performance of our algorithm, several examples are given as follows. Here we adopt the sampling density $l, k = 10$ (see Section 3.1) for all the examples.

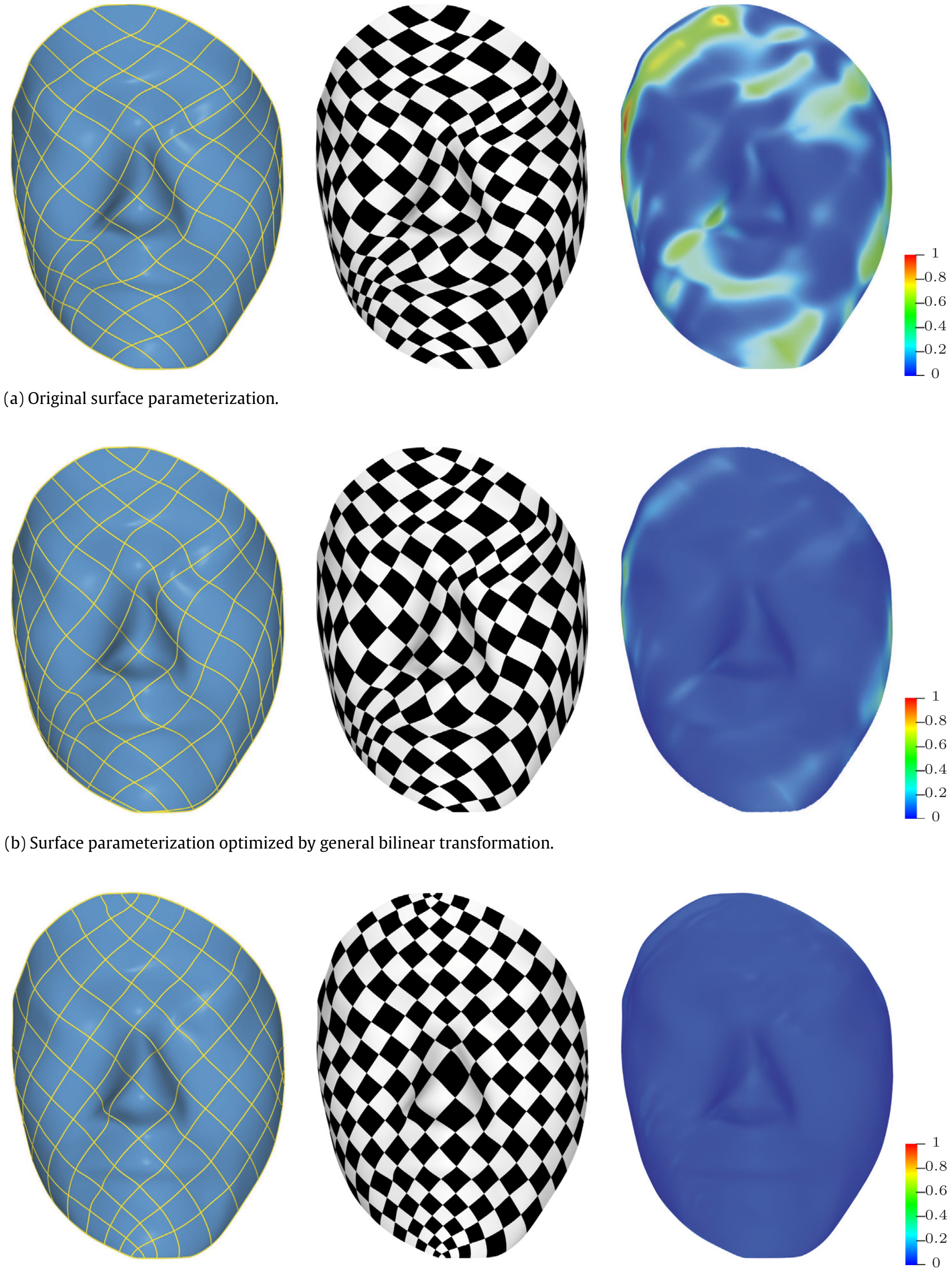
Fig. 3 illustrates the optimization result of a face NURBS surface with 43 by 43 control points. To show the performance, each surface is illustrated using three methods: iso-parametric curves, texture mapping and color-coded conformality. The iso-parametric curve network can also be seen as quad tessellations of the given freeform surfaces. Let $N(\delta, \zeta)$ denote the unit normal to the surface. In the color-coded images, we measure for a parameterization $\mathbf{X}(\delta, \zeta)$ the following conformality function across the surface

$$E(\delta, \zeta) = \left\| N(\delta, \zeta) \times \frac{\partial X}{\partial \delta}(\delta, \zeta) - \frac{\partial X}{\partial \zeta}(\delta, \zeta) \right\| ,$$

and use colors to illustrate the value of $E(\delta, \zeta)$ (blue is good conformality and red is bad conformality). The surface parameterization $\mathbf{X}(\delta, \zeta)$ is conformal at a point (δ, ζ) if and only if $E(\delta, \zeta) = 0$.

As shown in Fig. 3, the face NURBS surface is optimized using the general bilinear transformation and the freeform transformation, respectively. From Fig. 3, it can be seen that the original surface parameterization is far from being conformal. Texture mapping of the original parameterization involves a lot of shape distortion and color-coded image of the original parameterization includes a couple of green areas. By the Ricci flow method in [21], we compute the conformal mapping of its discretized mesh (see Fig. 4). By approximating the conformal mapping using the least square method, we obtain the initial freeform transformation with maximal conformality deviation 0.16, which is further optimized and refined until the conformal deviation is smaller than the given user tolerance $\epsilon_c = 0.05$. From the iso-parametric curves, the texture mapping and the color-coded conformality in Fig. 3, we can see that the conformality of the surface can be improved to some extent by the general bilinear transformation. To better illustrate the conformality deviation, the maximal conformality deviation of the original surface is normalized to 1 for all examples of this section. Though the maximal conformality deviation of the resultant surface optimized by the general bilinear transformation is reduced to 0.41, the texture mapping result still involves a lot of shape distortion and the color-coded image has several green areas near the boundary. Compared with the general bilinear transformation, the freeform transformation is much more powerful. By setting the conformality tolerance $\epsilon_c = 0.05$, we obtain the optimal freeform transformation with 17 by 17 control points. After the parameterization optimization by the freeform transformation, the texture mapping of the optimized parameterization well keeps the local shape and the color-coded conformality image of the optimized parameterization is almost blue.

Another example is given in Fig. 5, where a car top surface with 34 by 34 control points is optimized using the general bilinear transformation and the freeform transformation. The conformal mapping of its 3D discretized mesh, the final general bilinear transformation and the freeform transformation are illustrated in Fig. 6. From the results in Fig. 5, we can see that both the general bilinear transformation and the freeform transformation can improve the conformality of the given car top surface. By using the general bilinear transformation, the conformality deviation of the given car top surface is reduced to 0.17. However, the texture mapping result and the color-coded conformality image still involve large shape distortion and green areas near the boundaries of the tail, respectively. Compared with the limited ability of the general bilinear transformation, the maximal conformality deviation of the car top surface has been reduced to 0.09 by the initial freeform



(a) Original surface parameterization.

(b) Surface parameterization optimized by general bilinear transformation.

(c) Surface parameterization optimized by freeform transformation.

Fig. 3. Optimization of a face surface with 43 by 43 control points. (a) The iso-parametric curves, the texture mapping and the color-coded conformality of the original parameterization; (b) the iso-parametric curves, the texture mapping and the color-coded conformality of the parameterization optimized by the general bilinear transformation; (c) the iso-parametric curves, the texture mapping and the color-coded conformality of the parameterization optimized by the freeform transformation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

transformation, which is further optimized to the user specified tolerance $\epsilon_c = 0.02$ by the final freeform transformation with 32 by 32 control points. Both the texture mapping and the color-coded conformality images demonstrate the effectiveness of the freeform transformation.

Both the above face and car top examples have regular surface boundaries (almost a rectangle). To show the ability of our method to handle NURBS surfaces with irregular boundaries, a terrain NURBS surface is given in Fig. 7, in which we measure the data near the interesting mountain area and reconstruct the NURBS surface from its triangular mesh using a least square method in [9]. The conformal mapping of its 3D discretized mesh, the final general bilinear transformation and the freeform transformation are illustrated in Fig. 8. From Fig. 7, partially due to its irregular boundaries, the results of rendering and tessellation applications are far from being satisfied. Both the general bilinear transformation and freeform transformation methods are applied to optimize the conformality of the

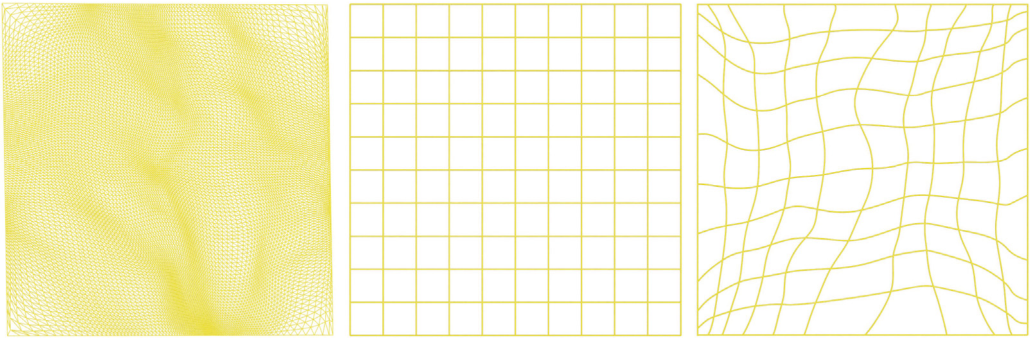
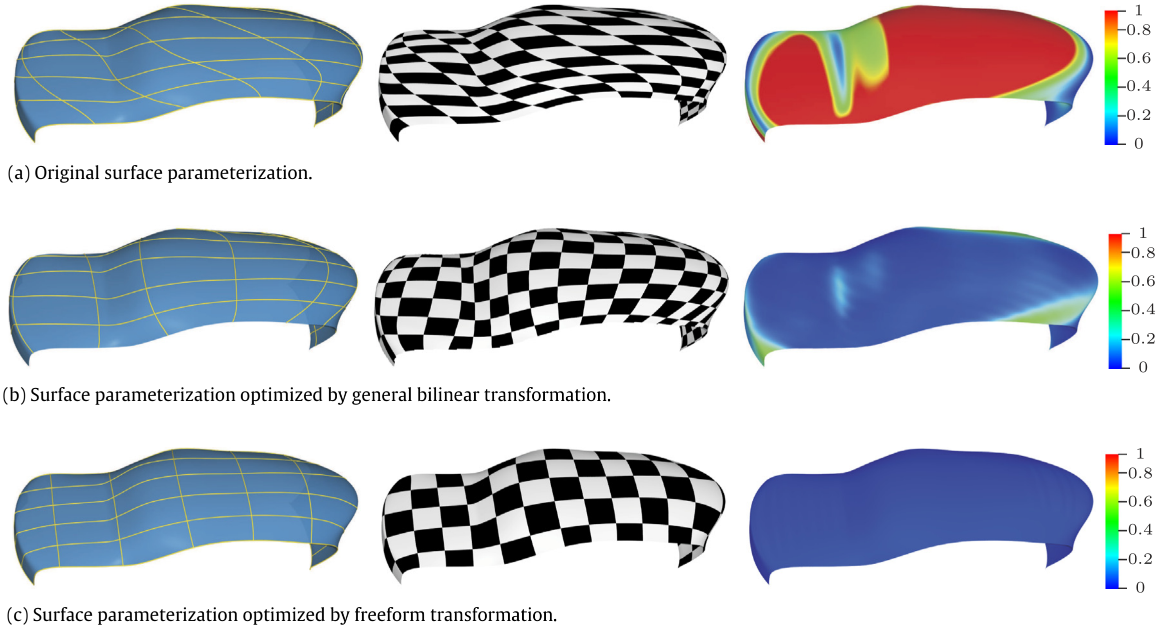


Fig. 4. The freeform transformation of the face NURBS surface. The conformal mapping of the 3D discrete mesh, the iso-parametric lines in the parameter domain of the final freeform transformation and the iso-parametric curves of the freeform transformation are arranged from left to right.



(a) Original surface parameterization.

(b) Surface parameterization optimized by general bilinear transformation.

(c) Surface parameterization optimized by freeform transformation.

Fig. 5. Optimization of a car top surface with 43 by 43 control points. (a) The iso-parametric curves, the texture mapping and the color-coded conformality of the original parameterization; (b) the iso-parametric curves, the texture mapping and the color-coded conformality of the parameterization optimized by the general bilinear transformation; (c) the iso-parametric curves, the texture mapping and the color-coded conformality of the parameterization optimized by the freeform transformation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

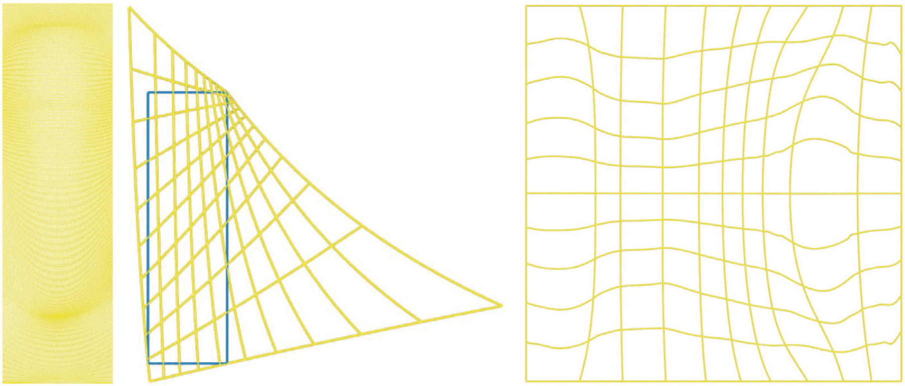
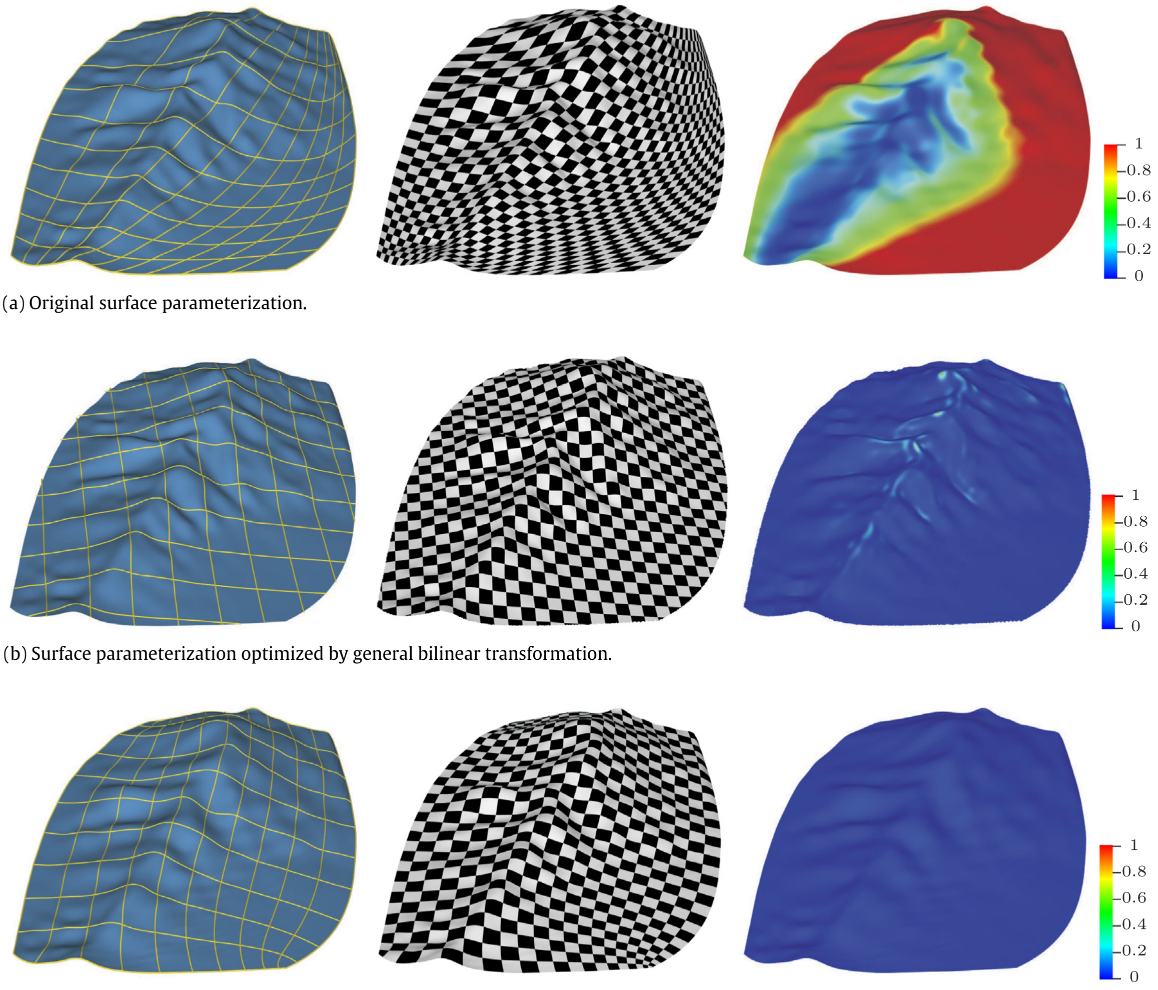


Fig. 6. The transformation of the car top NURBS surface. The conformal mapping of the 3D discrete mesh, the iso-parametric lines of the general bilinear transformation (the blue rectangle is the boundaries of the original parameter domain), and the iso-parametric curves of the freeform transformation are arranged from left to right. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

terrain surface. Though the conformality of the terrain surface is largely improved by the general bilinear transformation, the resultant color-coded image still has green areas along the mountain ridge. The freeform transformation exhibits to be more powerful and flexible than the general bilinear transformation. Especially, the freeform transformation improves the surface conformality while keeping the surface boundaries. Also to show the efficiency and effectiveness of our method to improve the confor-



(a) Original surface parameterization.

(b) Surface parameterization optimized by general bilinear transformation.

(c) Surface parameterization optimized by freeform transformation.

Fig. 7. Optimization of a terrain surface with 34 by 34 control points. (a) The iso-parametric curves, the texture mapping and the color-coded conformality of the original parameterization; (b) the iso-parametric curves, the texture mapping and the color-coded conformality of the parameterization optimized by the general bilinear transformation; (c) the iso-parametric curves, the texture mapping and the color-coded conformality of the parameterization optimized by the freeform transformation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

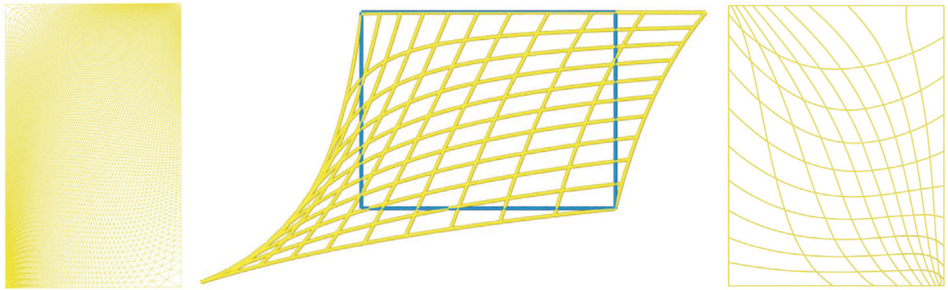


Fig. 8. The freeform transformation of the terrain NURBS surface. The conformal mapping of the 3D discrete mesh, the iso-parametric lines of the general bilinear transformation (the blue rectangle is the boundaries of the original parameter domain) and the iso-parametric curves of the freeform transformation are arranged from left to right. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

mality of complex and multiple freeform surfaces, an example of the bunny model with 316 patches is given in Fig. 9. The layout of the subsurface is illustrated in Fig. 9(a) and the conformality of the bunny model is improved by the general bilinear transformation and the freeform transformation. For the complex model with multiple patches, the parametric continuity between adjacent surfaces is combined with the conformality energy during the optimization procedure. The color-coded conformality images are illustrated in Fig. 9, from which it can be seen that the freeform transformation is superior to the general bilinear transformation at improving the surface conformality near the bunny neck.

The above four examples show that the freeform transformation can reduce the conformality deviation of given freeform surfaces to the user specified tolerance. Compared with the general bilinear transformation, the freeform transformation can improve the surface conformality while keeping its boundaries. Also, the time cost of bilinear transformation and the freeform transformation for the above four examples are given in Table 1. From Table 1, we can see that the parameterization quality of the resultant sur-

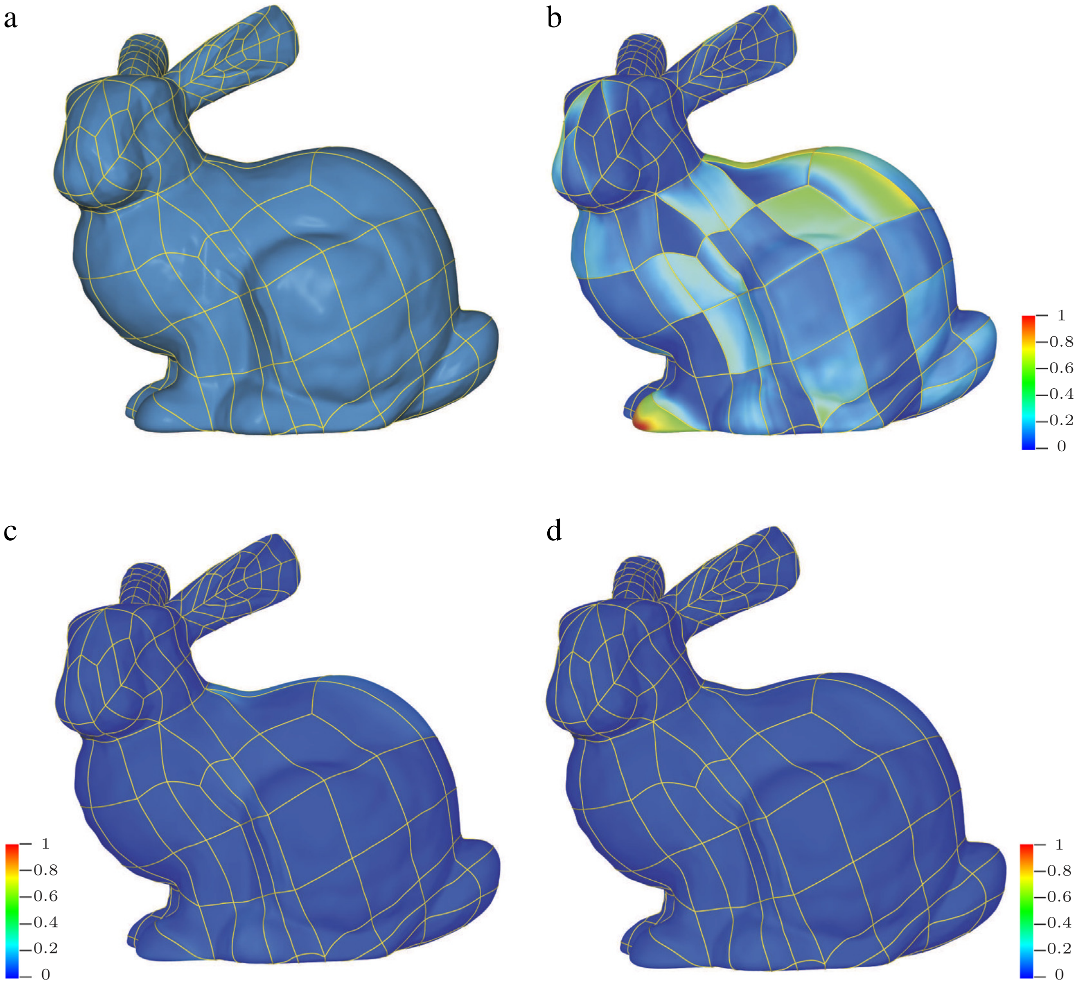


Fig. 9. The transformation of the bunny model: (a) the bunny model (the yellow curves are the subsurface boundaries); (b) the conformality of the original bunny model; (c) the conformality of the resultant model optimized by the general bilinear transformations; (d) the conformality of the resultant model optimized by the freeform transformations. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

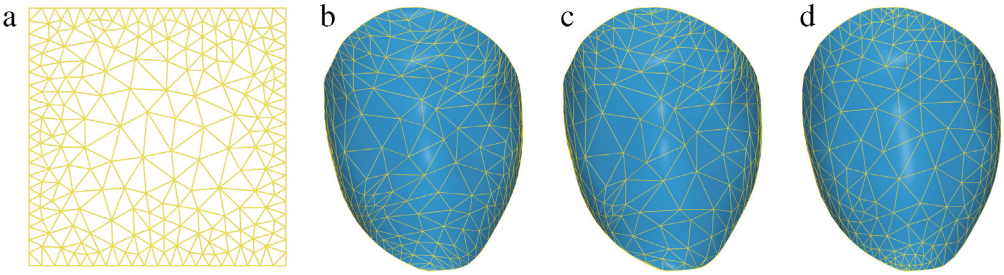


Fig. 10. Face triangulation. (a) 2D triangulation with minimal angle 33.9°; (b) the surface tessellation by mapping the 2D triangulation in (a) on the original surface with minimal angle 2.2°; (c) the surface tessellation by mapping the 2D triangulation in (a) on the optimized surface using the general bilinear transformation with minimal angle 4.1°; (d) the surface tessellation by mapping the 2D triangulation in (a) on the hierarchical freeform surface with minimal angle 20.3°.

Table 1
Time cost (ms) of reparameterization methods.

	Face	Cartop	Terrain	Bunny
General bilinear transformation	31	40	36	350
Freeform transformation	65	87	76	455

face is much better than the previous results in [40] while the time cost of the freeform transformation is comparable to that of the general bilinear transformation.

To show the efficiency of the hierarchical freeform surface for tessellation application, an example is given to tessellate the face surface in Fig. 3. The parameter domain is first triangulated using the constrained Delaunay triangulation algorithm in [47] by setting the minimal angle $\theta = 33.9°$. Then the 2D triangles are mapped to the original surface, the optimized surface by general bilinear transformation and the hierarchical freeform surface (see Fig. 10). From Fig. 10, the hierarchical freeform surface preserves the triangle angle better than the original surface and the optimized surface using the general bilinear transformation. We can see that the quality of the triangulation highly depends on the surface parameterization. Furthermore, we demonstrate the superiority of the hierarchical freeform surface to handle more complex trimmed NURBS surfaces. For trimmed NURBS surfaces, the tessellation method presented in [48] is implemented on the

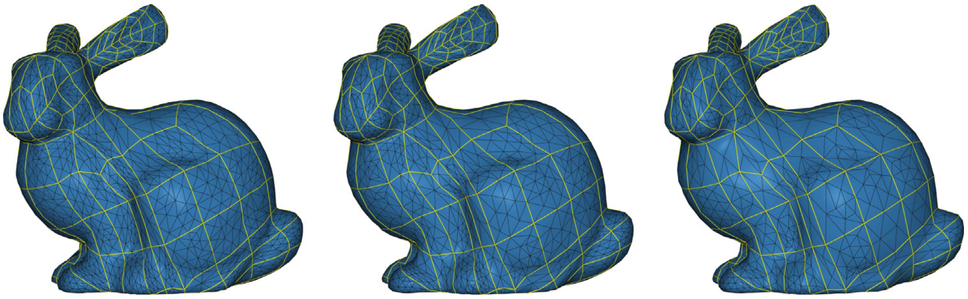


Fig. 11. Tessellation of the bunny model. Left: the original model; middle: the optimized surface by general bilinear transformation; right: the hierarchical freeform surface.

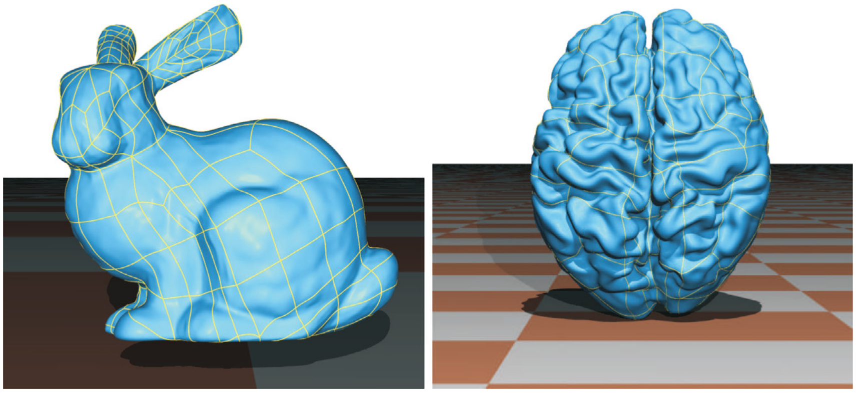


Fig. 12. Ray tracing of surfaces.

Table 2
Tessellation results of the bunny model with edge and face tolerances 0.001.

	Original model	General bilinear tran.	Hierarchical freeform model
Vertex number	3089	2389	1637
Face number	6172	4774	3270
Mesh deviation	0.00095	0.00096	0.00092
Minimal angle	3.9°	5.3°	11.6°
Time (ms)	35	43	47

original surface, the optimized surface by the general bilinear transformation and the hierarchical freeform surface, where the geometric approximation errors for the stitching boundaries and the surfaces are controlled under the given tolerances (see Fig. 11). The main advantage of the method in [48] is that it produces much fewer triangles than traditional methods [11], which is preferred for subsequent CAD applications. The trimming curves are first approximated by polylines under a prescribed distance tolerance, which is followed by the NURBS surface tessellation based on kd-tree subdivisions. The key step of the surface tessellation in [48] is to generate the constrained triangulation of the sampling point set of the trimming curves and surfaces. As the constrained triangulation of point clouds is a non-trivial problem, practically all NURBS tessellation algorithms generate the final triangulation in the parameter domain of the surface, which is reasonable as long as the surface does not deform the polygon too much. As the parameter domain of the hierarchical freeform surface better keeps the shape of the NURBS surface geometry (more conformal parameterization), less triangles will be introduced to tessellate the NURBS models under the specified tolerances, which is verified by the examples given in Fig. 11. The computation time, the number of triangles, the minimal angle and the mesh deviation are reported in Table 2. From Table 2, we can see that the hierarchical freeform surface will introduce less triangles than the original surface and the optimized surface by general bilinear transformation under the same subdivision tolerances, which demonstrates the superiority of the hierarchical freeform surface in surface tessellation application.

In computer graphics, ray tracing is a technique for generating an image by tracing the path of light through pixels in an image plane and simulating the effects of its encounters with virtual objects. The technique is capable of producing a very high degree of visual realism, usually higher than that of typical scanline rendering methods, but at a greater computational cost. The intersection between the ray tracing lines and the model accounts for most of the time cost. To show the efficiency of the hierarchical freeform surface for rendering, the bunny model in Fig. 9 and a brain model are rendered using the ray tracing method in [49] (see Fig. 12), where the NURBS surfaces are ray traced by intersection algorithms between rays and NURBS surfaces without the preprocess of tessellating the NURBS surfaces. Directly rendering free-form surfaces has several advantages compared with traditional methods, which convert the NURBS surfaces into triangles before rendering. The time-consuming triangulation preprocessing can be omitted and the accuracy is not limited to a certain distance from the observer. For the direct rendering of NURBS surfaces in [49], roughly 60% of the CPU time is spend in the surface evaluation, including the calculation of partial derivatives. Therein, a Newton iteration-based intersection approach is used for the intersection test between the rays and NURBS surfaces. The parameterization quality highly affects the convergence speed of the intersection algorithms and surfaces with better parameterization are advantageous for intersection computation due to smoother intersection curves and faster algorithm convergence, which has been verified and reported in [50]. For the original surface, the optimized surface by the general bilinear transformation and the hierarchical freeform surface, the time cost of the rendering algorithm in [49] is illustrated in Table 3. The rendering image resolution is set to 1600 × 1200. The bunny model is composed of 316 NURBS patches while the brain model has 202 NURBS patches. From Table 3, we can see that the superiority of hierarchical freeform surfaces to the general bilinear method is obvious for the rendering application.

Table 3
Rendering time cost (s) of ray tracing method.

	Bunny	Brain
Original surface	833	554
Optimized surface by general bilinear transformation	415	308
Hierarchical freeform surface	45	36

Besides the conformality, equiareality should also be considered in the freeform surface construction. In most cases, the freeform surfaces are not developable. Thus we cannot obtain both conformal and area preserving parameterizations. The conformality of the freeform surfaces is important for applications such as rendering (texture mapping), tessellation (keeping angles and shapes of the triangle), surface intersections and so on. Also the equiareality is important for sampling/tessellation applications (sampling density). There is a tradeoff between the conformality and equiareality of the freeform surfaces. In this paper, we focus on the conformality of the freeform surfaces. To improve the equiareality of the freeform surfaces is left as a future work. Also feature awareness of the surface parameterization is another important issue, which should be considered for the spline construction.

5. Conclusions

In order to improve conformality of given NURBS surfaces, an optimization algorithm is presented in this paper based on freeform transformations (hierarchical freeform surfaces). A nonlinear energy measuring the conformality deviations and its numerical approximation are formulated. To minimize the discretized version of the conformal energy, the initial freeform transformation is first obtained by approximating the conformal mapping of the 3D discretized mesh, which is computed using Ricci flow method. To further improve the conformality of the initial freeform transformation, the nonlinear conformality optimization and the transformation refinement interleave until the conformality deviation is reduced to the user specified tolerance. Several examples are given to show the performance of our algorithm for rendering and tessellation applications.

Acknowledgments

This work was supported by National Science Foundation (NSF) CCF-1544267, the China National Natural Science Foundation (61202146, 61272243, 61472224, and 61472225), Shandong Key Research and Development Program (2015GGX106006), the special funding of independent innovation and transformation of achievements in Shandong Province under Grant No. 2014ZZCX08201, the National High-Tech Research and Development Plan of China under Grant No. 2014AA01A302, and the special funds of Taishan scholar construction project, Young Scholars Program of Shandong University under Grant No. 2015WLJH41.

References

[1] Michael E, Martin K, Alexander S, Niloy JM, Helmut P, Mark P. Paneling architectural freeform surfaces. ACM Trans Graph 2010;29(4).

[2] Cheung G, Lau R, Li F. Incremental rendering of deforming NURBS surfaces. In: VRST 03: Proceedings of the ACM symposium on Virtual reality software and technology. New York, NY, USA: ACM Press; 2003. p. 48–55.

[3] Chuang JH, Lin CH, Hwang WC. Variable-radius blending of parametric surfaces. Vis Comput 1995;11(10):513–25.

[4] Chuang JH, Hwang WC. Variable-radius blending by constrained spine generation. Vis Comput 1997;13(7):316–29.

[5] Hamann B, Tsai PY. A tessellation algorithm for the representation of trimmed nurbs surfaces with arbitrary trimming curves. Comput-Aided Des 1996; 28(6/7):461–72.

[6] Kumar S, Manocha D. Efficient rendering of trimmed NURBS surfaces. Comput-Aided Des 1995;27(7):509–21.

[7] Ng WMM, Tan ST. Incremental tessellation of trimmed parametric surfaces. Comput-Aided Des 2000;32(4):279–94.

[8] Piegl LA, Richard AM. Tessellating trimmed NURBS surfaces. Comput-Aided Des 1995;27(1):15–26.

[9] Piegl LA, Tiller W. The NURBS book. 2nd ed. New York: Springer; 1997.

[10] Piegl LA, Tiller W. Filling n-sided regions with NURBS patches. Vis Comput 1999;15(2):77–89.

[11] William LL. Tessellation of trimmed NURB surfaces. Comput Aided Geom Design 1996;13(2):163–77.

[12] Yang YJ, Yong JH, Zhang H, Paul JC, Sun JG. A rational extension of Piegl's method for filling n-sided holes. Comput-Aided Des 2006;38(11):1166–78.

[13] Yang YJ, Zeng W, Zhang H, Paul JC, Yong JH. Projection of curves on b-spline surfaces using quadratic reparameterization. J Graph Models 2010; 72(5):47–59.

[14] Yang YJ, Zeng W, Yang CL, Meng XX, Yong JH, Deng BL. G1 continuous approximate curves on NURBS surfaces. Comput-Aided Des 2012;44(9): 824–34.

[15] Lee ETY, Lucian ML. Möbius reparametrizations of rational B-splines. Comput Aided Geom Design 1991;8(3):213–5.

[16] Floater MS, Hormann K. Surface parameterization: a tutorial and survey. In: Dodgson NA, Floater MS, Sabin MA, editors. Advances in multiresolution for geometric modelling. Mathematics and visualization, Berlin, Heidelberg: Springer; 2005. p. 157–86.

[17] Ray N, Li WC, Levy B, Sheffer A, Alliez P. Periodic global parameterization. ACM Trans Graph 2006;25(4):1460–85.

[18] Zou G, Hu JX, Gu XF, Hua J. Authalic parameterization of general surfaces using Lie advection. IEEE Trans Vis Comput Graphics 2011;17(12):2005–14.

[19] Hormann K, Lévy B, Sheffer A. Mesh parameterization: Theory and practice. In: SIGGRAPH 2007 course notes, 2007.

[20] Sheffer A, Praun E, Rose K. Mesh parameterization methods and their applications. Found Trends Comput Graph Vis 2006;2(2):105–71.

[21] Gu DX, Zeng W, Luo F, Yau ST. Numerical computation of surface conformal mappings. Comput Methods Funct Theory (CMFT) 2011;11(2):747C787.

[22] Lévy B, Petitjean S, Ray N, Maillot J. Least squares conformal maps for automatic texture atlas generation. ACM Trans Graph 2002;21(3):362–71. Proceedings of SIGGRAPH 2002.

[23] Mullen P, Tong YY, Alliez P, Desbrun M. Spectral conformal parameterization. Comput Graph Forum 2008;27(5):1487–94.

[24] Costantini P, Farouki RT, Manni C, Sestini A. Computation of optimal composite re-parameterizations. Comput Aided Geom Design 2001;18(9):875–97.

[25] Farouki RT. Optimal parameterizations. Comput Aided Geom Design 1997; 14(2):153–68.

[26] He L, Schaefer S, Hormann K. Parameterizing subdivision surfaces. ACM Trans Graph 2010;29(4).

[27] He L, Loop C, Schaefer S. Improving the parameterization of approximate subdivision surfaces. Comput Graph Forum 2012;31(7):2127–34.

[28] Jüttler B. A vegetarian approach to optimal parameterizations. Comput Aided Geom Design 1997;14(9):887–90.

[29] Ong BH. An extraction of almost arc-length parameterization from parametric curves. Ann Numer Math 1996;3:305–16.

[30] Shpitalni M, Koren Y, Lo CC. Realtime curve interpolators. Comput-Aided Des 1994;26(11):832–8.

[31] Wang FC, Yang DCH. Nearly arc-length parameterized quintic spline interpolation for precision machining. Comput-Aided Des 1993;25(5):281–8.

[32] Wang FC, Wright PK, Barsky BA, Yang DCH. Approximately arc-length parameterized C^3 quintic interpolatory splines. ASME J Mech Des 1999;121: 430–9.

[33] Wever U. Optimal parameterization for cubic spline. Comput-Aided Des 1991; 23(9):641–4.

[34] Yang DCH, Kong T. Parametric interpolator versus linear interpolator for precision CNC machining. Comput-Aided Des 1994;26(3):225–34.

[35] Yang DCH, Wang FC. A quintic spline interpolator for motion command generation of computer-controlled machines. ASME J Mech Des 1994;116: 226–31.

[36] Yeh SS, Hsu PL. The speed-controlled interpolator for machining parametric curves. Comput-Aided Des 1999;31(5):349–57.

[37] Yang YJ, Yong JH, Zhang H, Paul JC, Sun JG. Optimal parameterizations of Bézier surfaces. In: 2nd international symposium on visual computing, ISVC06, 2006, p. 672–81.

[38] Farin G. NURBS from projective geometry to practical use. 2nd ed. A K Peters; 1999.

[39] Yang YJ, Zeng W, Yang CL, Deng BL, Meng XX, Iyengar SS. An algorithm to improve parameterizations of rational Bezier surfaces using rational bilinear reparameterization. Comput-Aided Des 2013;45(3):628–38.

[40] Yang YJ, Zeng W, Song TQ. Optimizing conformality of NURBS surfaces by general bilinear transformations. Comput-Aided Des 2015;46(6):12–25.

[41] Do Carmo MP. Differential geometry of curves and surfaces. Prentice Hall; 1976.

[42] Richard LB, John DF. Numerical analysis. Cengage Learning; 2010.

[43] Kanzow C, Yamashita N, Fukushima M. Levenberg-marquardt methods with strong local convergence properties for solving nonlinear equations with convex constraints. J Comput Appl Math 2004;173(2):321–43.

[44] Horst R, Pardalos PM, Thoai NV. Introduction to global optimization. 2nd ed. Springer; 2000.

[45] Deng B, Pottmann H, Wallner J. Functional webs for freeform architecture. Comput Graph Forum 2011;30(5):1369–78.

[46] Liu Y, Pottmann H, Wallner J, Yang YL, Wang W. Geometric modeling with conical meshes and developable surfaces. ACM Trans Graph 2006;25(3): 681–9.

[47] Shewchuk JR. Delaunay refinement algorithms for triangular mesh generation. Comput Geom Theory Appl 2002;22(1–3):21–74.

[48] Balzs A, Guthe M, Klein R. Efficient trimmed NURBS tessellation. J WSCG 2004; 12(1):27–33.

[49] Abert O, Geimer M, Muller S. Direct and fast ray tracing of NURBS. In: IEEE symposium on surfaces interactive ray tracing, 2006, p. 161–8.

[50] Casciola G, Morigi S. Reparameterization of NURBS curves. Int J Shape Model 1996;2(2–3):103–16.