Automatic Fairing of Two-Parameter Rational B-Spline Motion

Anurag Purwar
e-mail: anurag.purwar@stonybrook.edu

Xiaoyi Chi

Qiaode Jeffrey Ge
e-mail: qiaode.ge@stonybrook.edu

Department of Mechanical Engineering,
Stony Brook University,
Stony Brook, NY 11794-2300

This paper deals with the problem of automatic fairing (or fine-tuning) of two-parameter rational B-spline spherical and spatial motions. The results presented in this paper extend the previous results on fine-tuning of one-parameter rational B-spline motions. A dual quaternion representation of spatial displacements is employed and the problem of fairing two-parameter motions is studied as a surface fairing problem in the space of dual quaternions. By combining surface fairing techniques from the field of computer aided geometric design with the computer aided synthesis of freeform rational motions, smoother (C^3 continuous) two-parameter rational B-spline motions are generated. Several examples are presented to illustrate the effectiveness of the proposed method. Techniques for motion smoothing have important applications in the Cartesian motion planning, camera motion synthesis, and spatial navigation in virtual reality systems. In particular, smoother two-parameter freeform motions have applications in the development of a kinematic based approach to geometric shape design and in five-axis NC tool path planning. [DOI: 10.1115/1.2803253]

1 Introduction

There has been a great deal of research on the problem of freeform motion synthesis, which is concerned with the development of freeform motions such as rational B-spline motions that approximate or interpolate a given set of displacements of an object. Quaternion representation of rotation (Bottema and Roth [1]) is widely recognized to be an effective way of handling the interpolation of rotations (Shoemake [2], Nielson and Heiland [3], Kim and Nam [4], Kim et al. [5]). Ge and Ravani [6,7] used dual quaternions to deal with the interpolation of spatial displacements that include both rotations and translations. Purwar and Ge [8] used dual numbers as the weights to wrest finer control over the synthesis of rational B-spline motions. Juttler and Wagner [9] provided a rich set of tools and algorithms for motion design, although their choice of representation of spatial displacements was based on treating translation and rotation separately. See Roschel [10] and Purwar [11] for a comprehensive list of references on the topic of computer aided motion synthesis.

Despite rapid advances in the synthesis of rational motions, there has been much less research on the issue of fine-tuning a rational motion to achieve desired smoothness. Srinivasan and Ge [12] presented algorithms for fine-tuning both the path and the speed of a rational B-spline motion. While the path-smoothing algorithm automatically removes the third order discontinuities in the path of the motion, the speed smoothing algorithm results in a B-spline motion that keeps its kinetic energy nearly constant. In the intervening years, other researchers have also explored various techniques to smooth out the motion—Fang et al. [13] used a lowpass filter coupled with an adaptive, mediative filter for angular velocities to achieve smooth rotation; Lee and Shin [14] used an algorithm that iteratively minimizes the energy function reflecting the forces and torques, exerted on a moving object; Kim et al. [15] improved upon the previous algorithm by minimizing the weighted sum of strain energy and the sum of square errors (SSE). There is a wealth of signal processing methods available for noise removal but due to the nonlinear nature of orientation space, they are not directly applicable. Lee and Shin [16] took this approach by first transforming the orientation data into their counterparts in a vector space, and then applied a convolution filter to it before transforming it back to the orientation space. More recently, wavelet based methods as well as genetic algorithms have been developed for motion fine-tuning (Hsieh [17,18], Hsieh and Chang [19]).

All the aforementioned works deal only with one-parameter motions. Ge and Sirchia [20] proposed the concept of two-parameter rational B-spline motions, which have potential applications in the development of a kinematics based approach to shape generation (see Ge [21]). In difference to one-parameter rational motions, the point trajectory of a two-parameter rational motion traces out a rational surface instead of a rational curve. This has the potential to reduce some of the freeform surface generation problems in CNC tool path planning into that of fine-tuning a two-parameter rational B-spline motion such that the resulting swept surface best approximates the desired freeform surface. In this paper, for the first time, we have brought together kinematic geometry of two-parameter B-spline motions, computer aided geometric design, and the fairing of surfaces (see Kjellander [22], Nowacki [23], Lott and Pullin [24] for fairing of surfaces) to develop an innovative method for fine-tuning of two-parameter B-spline motions. With the aid of dual quaternions, we represent a two-parameter rational B-spline rational motion as a surface in the space of dual quaternions. We then employ an improved surface fairing technique proposed by Hahmann [25] for fine-tuning of the dual quaternion surface. This results in an algorithm that automatically detects and removes third order discontinuity in the two-parameter B-spline motion.

The rest of the paper is organized as follows. Section 2 gives a review on the quaternion and dual quaternion representation of spherical and spatial displacements, respectively. Section 3 shows how a two-parameter rational B-spline motion can be represented by a surface in the space of dual quaternions. Section 4 reviews some of the surface fairing techniques including the fairness criteria. Sections 5 and 6 present the algorithms for fine-tuning of two-parameter rational B-spline spherical and spatial motions, respectively, along with a few examples.

Contributed by the Mechanisms and Robotics Committee of ASME for publication in the Journal of Mechanical Design. Manuscript received October 11, 2006; final manuscript received January 16, 2007; published online December 7, 2007. Review conducted by Hashem Ashrafiuon. Paper presented at the ASME 2005 Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2005), Long Beach, CA, September 24–28, 2005.

2 Representation of Spatial Displacements

In this section, we review the concepts of quaternions and dual quaternions insofar as necessary for the development of the current paper.

2.1 Unit Quaternion as Rotation. Any rotation in three-dimensional space has a rotation axis and a rotation angle about this axis. Let $\mathbf{s}=(s_x,s_y,s_z)$ denote a unit vector along the axis and θ denote the angle of rotation. They can be used to define the so-called *Euler–Rodrigues parameters:*

$$q_1 = s_x \sin(\theta/2), \quad q_2 = s_y \sin(\theta/2), \quad q_3 = s_z \sin(\theta/2), \quad q_4 = \cos(\theta/2) \tag{1}$$

The Euler–Rodrigues parameters and the quaternion units, 1, $\mathbf{i},\mathbf{j},\mathbf{k}$ can be combined to define a quaternion of rotation:

$$\mathbf{Q} = q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k} + q_4. \tag{2}$$

A quaternion $\mathbf{Q}$, at times, is also written as an ordered quadruple (q_1,q_2,q_3,q_4). The reader is advised to refer to Bottema and Roth [1] and McCarthy [26] for details on quaternion algebra.

Let $\mathbf{x}=(x_1,x_2,x_3,x_4)$, and $\mathbf{X}=(X_1,X_2,X_3,X_4)$ denote homogeneous coordinates of a point before and after a rotation, respectively. They can be assembled as quaternions:

$$\mathbf{x} = x_1\mathbf{i} + x_2\mathbf{j} + x_3\mathbf{k} + x_4, \quad \mathbf{X} = X_1\mathbf{i} + X_2\mathbf{j} + X_3\mathbf{k} + X_4$$

The coordinate transformation that describes the rotation of a point from $\mathbf{x}$ to $\mathbf{X}$ is given by the quaternion product:

$$\mathbf{X} = \mathbf{Q}\ \mathbf{x}\ \mathbf{Q}^{-1} \tag{3}$$

where $\mathbf{Q}^{-1}$ denotes the inverse of $\mathbf{Q}$ and is given by

$$\mathbf{Q}^{-1} = (1/S^2)(q_4 - q_1\mathbf{i} - q_2\mathbf{j} - q_3\mathbf{k})$$

with

$$S^2 = q_1^2 + q_2^2 + q_3^2 + q_4^2$$

When $S^2=1$, a quaternion $\mathbf{Q}$ is called a unit quaternion and its inverse $\mathbf{Q}^{-1}$ equals to its conjugate $\mathbf{Q}^*=q_4-q_1\mathbf{i}-q_2\mathbf{j}-q_3\mathbf{k}$.

We can relate the quaternion representation given by Eq. (3) to the more familiar matrix form as

$$\mathbf{X} = [R]\mathbf{x}$$

where $[R]$ is an orthogonal matrix given by

$$[R] = \frac{1}{S^2}\begin{bmatrix} q_4^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_4q_3) & 2(q_1q_3+q_4q_2) \\ 2(q_2q_1+q_4q_3) & q_4^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3-q_4q_1) \\ 2(q_3q_1-q_4q_2) & 2(q_3q_2+q_4q_1) & q_4^2-q_1^2-q_2^2+q_3^2 \end{bmatrix} \tag{4}$$

The quaternion components of $\mathbf{Q}$ can be considered as homogeneous coordinates of a rotation since multiplying them by a scalar leaves the rotation matrix $[R]$ invariant. Ravani and Roth [27] considered these components as defining a point in a projective three space, called the *Image Space* of spherical kinematics. In this way, a polynomial curve in the Image Space correspond to a rational motion. By applying computer aided geometric design (CAGD) techniques for designing curves in the Image Space, we obtain rational motions in the Cartesian space.

2.2 Dual Quaternion as Spatial Displacement. For a general displacement in the Cartesian space E^3, Eq. (3) can be extended naturally to accommodate translation as follows:

$$\mathbf{X} = \mathbf{Q}\ \mathbf{x}\ \mathbf{Q}^{-1} + \mathbf{d} \tag{5}$$

where $\mathbf{d}=d_1\mathbf{i}+d_2\mathbf{j}+d_3\mathbf{k}$ is a vector quaternion corresponding to the translation component, which is defined by the nonhomogeneous parameters (d_1,d_2,d_3).

A spatial displacement can be elegantly represented by combining rotation with the translation using dual quaternions (Bottema and Roth [1], McCarthy [26]). A dual quaternion is given by

$$\hat{\mathbf{Q}} = \mathbf{Q} + \varepsilon\mathbf{R} \tag{6}$$

where ε is the dual unit with the property $\varepsilon^2=0$, and $\mathbf{Q}$ and $\mathbf{R}$ are called real and dual parts of the dual quaternion $(\hat{\mathbf{Q}})$, respectively. The real part, $\mathbf{Q}=(Q_1,Q_2,Q_3,Q_4)$, is the quaternion of rotation as mentioned before and $\mathbf{R}=(R_1,R_2,R_3,R_4)$ is another quaternion whose components are given by

$$\begin{bmatrix} R_1 \\ R_2 \\ R_3 \\ R_4 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 0 & -d_3 & d_2 & d_1 \\ d_3 & 0 & -d_1 & d_2 \\ -d_2 & d_1 & 0 & d_3 \\ -d_1 & -d_2 & -d_3 & 0 \end{bmatrix}\begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \end{bmatrix} \tag{7}$$

The translation vector $\mathbf{d}$ (d_1,d_2,d_3) can be recovered from Eq. (7) in terms of $(\mathbf{Q},\mathbf{R})$ by using the following:

$$\mathbf{d} = -\frac{2}{S^2}\begin{bmatrix} R_4Q_1 - R_1Q_4 + R_2Q_3 - R_3Q_2 \\ R_4Q_2 - R_2Q_4 + R_3Q_1 - R_1Q_3 \\ R_4Q_3 - R_3Q_4 + R_1Q_2 - R_2Q_1 \end{bmatrix} \tag{8}$$

where $S^2=Q_1^2+Q_2^2+Q_3^2+Q_4^2$.

It is instructive to note here that $(\mathbf{Q},\mathbf{R})$ serve as homogeneous coordinates of spatial displacements since multiplying each one of them by a nonzero scalar or a dual number yields the same rotation matrix and translation vector $\mathbf{d}$. Ravani and Roth [27] considered $\hat{\mathbf{Q}}=(\hat{Q}_1,\hat{Q}_2,\hat{Q}_3,\hat{Q}_4);\hat{Q}_i=Q_i+\epsilon R_i$ as a set of four homogeneous dual coordinates that define a point in the space of dual quaternions, called the Image Space of spatial displacements. A curve in the Image Space corresponds to a one-parameter motion in Cartesian space while a surface corresponds to a two-parameter motion. We develop the notion of two-parameter motions further in the next section.

3 Two-Parameter Rational B-Spline Motion

The concept of two-parameter freeform motions was developed by Ge and Sirchia [20] by combining the notion of the analytically determined two-parameter motions in theoretical kinematics with the concept of freeform surfaces in the field of CAGD. In this section, we briefly review that concept.

Given an $(n+1)\times(m+1)$ array of dual quaternions $\hat{\mathbf{Q}}_{i,j}(=\mathbf{Q}_{i,j}+\varepsilon\mathbf{R}_{i,j})$, we may define the following two tensor-product B-spline surfaces in the space of dual quaternions (or the Image Space), one each for the real and the dual part:

$$\mathbf{Q}^{n,m}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m}\mathbf{Q}_{i,j}N_{i,p}(u)N_{j,q}(v) \tag{9}$$

and

$$\mathbf{R}^{n,m}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m}\mathbf{R}_{i,j}N_{i,p}(u)N_{j,q}(v) \tag{10}$$

Assembling the real and the dual part from Eqs. (9) and (10), we define the dual quaternion surface:

$$\hat{\mathbf{Q}}^{n,m}(u,v) = \mathbf{Q}^{n,m}(u,v) + \varepsilon\mathbf{R}^{n,m}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m}\hat{\mathbf{Q}}_{i,j}N_{i,p}(u)N_{j,q}(v) \tag{11}$$

which represents the set of positions and orientations of an object that belong to a two-parameter B-spline motion. The bidirectional net of the $\hat{\mathbf{Q}}_{i,j}$ is called the *control net* of the B-spline image surface. $N_{i,p}(u)$ and $N_{j,q}(v)$ are the nonrational B-spline basis functions of degree $p-1$ and $q-1$, respectively, and defined on the knot vectors $(\mathbf{U};\mathbf{V})$:

$$\mathbf{U} = \{u_0 \le u_1 \cdots \le u_{n+p}\}, \quad \mathbf{V} = \{v_0 \le v_1 \cdots \le v_{m+q}\}$$

The above description is similar to that of a tensor product B-spline surface (see Farin [28] or Piegl and Tiller [29]) given as

$$\mathbf{X}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} \mathbf{b}_{i,j} N_{i,p}(u) N_{j,q}(v) \tag{12}$$

where given De Boor control points $\mathbf{b}_{i,j}$ are replaced by given dual quaternions $\hat{\mathbf{Q}}_{i,j}$.

Two-parameter B-spline motions enjoy properties, such as coordinate-frame invariance, convex hull property, boundedness, and local control, the latter three of which are due to the nature of B-spline basis functions (Schumaker [30], De Boor [31], or Piegl and Tiller [29]). See Ge and Sirchia [20] for a discussion on these properties.

4 Surface Fairing Techniques

As mentioned earlier, a two-parameter rational B-spline motion is represented by a B-spline surface in the space of dual quaternions. Thus, the problem of fine-tuning a two-parameter motion can be construed as a surface fairing problem in the same space. In CAGD, the surface is modeled as patches joined at boundaries. The continuity, characterized as the rth order derivative continuity (C^r), of the patches at the boundaries determines the smoothness of the surface. The goal of fine-tuning two-parameter motions is to examine and improve the order of the continuity at the patch boundaries of the corresponding dual-quaternion surfaces. In this section, we review some of the surface fairing techniques with an emphasis on the algorithm presented by Hahmann [25] for fairing bicubic B-spline surfaces. It is this algorithm that we will adapt to the space of dual quaternions for the fine-tuning of two-parameter rational B-spline motions.

Surfaces can be faired using *fairness constraints* or by performing *postprocessing fairing*. The former method usually incorporates a linearized physical based fairness criterion in the interpolation or approximation. These methods are usually nonlinear in nature and seek to create fair shapes by minimizing the energy of a curve or surface under given constraints (Welch and Witkin [32]). Wesselink and Veltcamp [33] present variational modeling techniques and tools for curves and surfaces. Greiner and Seidel [34] discuss approaches for energy functionals used in minimization methods. These methods produce fair and pleasant shapes but due to a global smoothing process the local control cannot be achieved.

The second method tries to remove unwanted surface wiggles by smoothing the control net, which defines the surface (Kjellander [22]). It is a variation of this method that we employ in this paper. Our method is based on Hahmann's [25] *automatic*, *local*, and *efficient* fairing method for bicubic B-spline surfaces. This method increases locally the smoothness of the surface from C^2 to C^3. Desirability of C^3 continuity stems from the fact that by reducing the difference in the third derivatives across a data point, the energy of a physical spline is decreased (Kjellander [22]). Hahmann's algorithm [25] is also flexible since it allows for different fairing steps, permits localized fairing process, and can impose tolerance control. In what follows, we review Hahmann's algorithm [25] to the extent necessary for the development of this paper.

A parametric tensor product B-spline surface $\mathbf{X}$ of degree (3;3) is defined by

$$\mathbf{X}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} \mathbf{b}_{i,j} N_{i,4}(u) N_{j,4}(v) \tag{13}$$

where

$$(u;v) \in \Omega := [u_3;u_{n+1}] \times [v_3;v_{m+1}]$$

and

$$u_i < u_{i+4} \quad (i=0,1,\ldots,n) \quad v_j < v_{j+4} \quad (j=0,1,\ldots,m)$$

The knots $u_4,\ldots,u_n$, $v_4,\ldots,v_m$ are called interior knots and are assumed to be of unit multiplicity. Hahmann [25] calls these inner knots as *free inner knots* since they are the potential sites for fairing process. A bicubic B-spline surface is at least C^2 continuous at the knots. The idea is to elevate the continuity at the joints (or inner knots) of the surface from C^2 to C^3 by modifying the control net. The algorithm proceeds by first defining a local fairness measure and then selecting the inner knot where the surface is least faired by the measure. Further, using a constrained least-squares approximation, the control net in the vicinity of the chosen knot is modified locally, which results in increasing the smoothness of the surface at the knot from C^2 to C^3. This process is repeated until a global measure of the fairness falls below a predefined tolerance level.

4.1 Fairing Criteria. Hahmann [25] defines a fairing criterion as follows: A B-spline surface $\mathbf{X}(u,v)$ of C^2 continuity is fairer at a free inner knot $[u_k,v_l]$, if $\mathbf{X}$ is C^3 at $[u_k,v_l]$.

Fortunately, bicubic B-spline surfaces have all mixed third order partial derivatives (i.e., $\mathbf{X}_{u^\nu v^\mu} = (\partial^{\nu+\mu}X)/\partial u^\nu \partial v^\mu$ with $\nu,\mu \geqslant 1$) continuous on the parametric domain $\Omega := [u_i,u_{i+4}] \times [v_j,v_{j+4}]$ since a B-spline surface is at least C^2 in either direction.

The sum of the square of the difference of $\mathbf{X}_{uuu}$ in the u direction and $\mathbf{X}_{vvv}$ in the v-direction at the interior knot $[u_k,v_l]$ is therefore an appropriate local fairness measure. Based on this premise, Hahmann [25] defines *discontinuity vectors*:

$$\begin{aligned}
\Delta_{uuu}(u_k,v_l) &= \mathbf{X}_{uuu}(u_k^-,v_l) - \mathbf{X}_{uuu}(u_k^+,v_l) \\
&= \sum_{j=0}^{m} \mathbf{b}_{k-1,j}^{(3,0)} N_{j,4}(v_l) - \sum_{j=0}^{m} \mathbf{b}_{k,j}^{(3,0)} N_{j,4}(v_l) \\
&= \sum_{i=k-4}^{k}\sum_{j=l-3}^{l-1} \alpha_{ij}\mathbf{b}_{ij} \\
\Delta_{vvv}(u_k,v_l) &= \mathbf{X}_{vvv}(u_k,v_l^-) - \mathbf{X}_{vvv}(u_k,v_l^+) \\
&= \sum_{i=0}^{n} \mathbf{b}_{i,l-1}^{(0,3)} N_{i,4}(u_k) - \sum_{i=0}^{n} \mathbf{b}_{i,l}^{(0,3)} N_{i,4}(u_k) \\
&= \sum_{i=k-3}^{k-1}\sum_{j=l-4}^{l} \beta_{ij}\mathbf{b}_{ij}
\end{aligned} \tag{14}$$

with

$$\alpha_{ij} = \alpha_{ij}(u_k,v_l) \text{ and } \beta_{ij} = \beta_{ij}(u_k,v_l)$$

For uniform knots, values of the coefficients α_{ij} and β_{ij} can be easily calculated and are given in Hahmann [25]. For details on the calculation of $\mathbf{X}_{uuu}$ and $\mathbf{X}_{vvv}$, see Hoschek and Lasser [35].

The local fairness measure L_{kl} at the point (u_k,v_l) is defined as

$$L_{kl} = \|\Delta_{uuu}(u_k,v_l)\|^2 + \|\Delta_{vvv}(u_k,v_l)\|^2 \tag{15}$$

The whole surface $\mathbf{X}$ can now be associated with a *global fairness measure*:

$$G_X = \sum_{(k,l)\in I} L_{kl} \tag{16}$$

where I denotes the index set for the knot pairs $(u_k;v_l)$ of the interior knots, $(k;l) \in I = (4;4);(4;5)\cdots(n;m)$. Global fairness measure given by Eq. (16) provides the stopping condition for an iterative procedure, to be described later.

Now that fairness measures have been prescribed, we can provide a sketch of the algorithm of Hahmann [25]. First, the local fairness measure L_{kl} is calculated from Eq. (15) for all interior knots. The knot with the maximum value of L_{ij} over all $(i,j) \in I$ is chosen as the site for applying local fairing step. Let us say this knot is (u_k, v_l). Then, after applying a fairing step, L_{kl} should be equal to zero, which translates into the conditions:

$$\Delta_{vvv}(u_k, v_l) = 0 \quad \text{and } \Delta_{uuu}(u_k, v_l) = 0$$

Imposing the above two conditions changes the control points in the vicinity of (u_k, v_l) and increases the continuity at the knot (u_k, v_l) from C^2 to C^3. However, it is also desirable that the control net, and thus in turn, the surface should deform as little as possible. Hahmann [25] sets up this problem by defining an objective function that seeks to minimize the changes to the control net in a small area. The problem is solved by using a classic least-squares approximation method with constraints:

$$\text{Min } F(\hat{\mathbf{b}}_{ij}) = \sum_{i=k-3}^{k-1} \sum_{j=l-3}^{l-1} \|\mathbf{b}_{ij} - \hat{\mathbf{b}}_{ij}\|^2 \tag{17}$$

subject to $\Delta_{vvv}=0$, $\Delta_{uuu}=0$, and where $\mathbf{b}_{ij}$ and $\hat{\mathbf{b}}_{ij}$ denote control points before and after an update, respectively. Even though discontinuity vectors involve more than nine control points, Eq. (17) indicates that only nine innermost points are chosen for an update at every step of fairing. Hahmann [25] argues that this is done (1) to obviate the need to solve a nonlinear problem and (2) to get a fairing step, which is as local as possible without restricting the solution too much.

The Lagrange multipliers method is used to solve this problem:

$$\Phi(\hat{\mathbf{b}}_{ij}, \lambda, \mu) = F(\hat{\mathbf{b}}_{ij}) + \lambda(\Delta_{uuu}(u_k, v_l)) + \mu(\Delta_{vvv}(u_k, v_l)) \rightarrow \min \tag{18}$$

In this paper, the isophote technique (Poeschl [36]) is used for obtaining a visual feedback on the fairness of the quaternion surface[1] imperfections. A fair surface exhibits smooth and regular isophotes. Connecting all points of a surface along which the angle between the light direction and the surface normal is constant and in the range between −90 deg and 90 deg, we obtain a line of constant illumination intensity called isophote. Only curves with angles between 0 deg and 90 deg are in the illumination part.

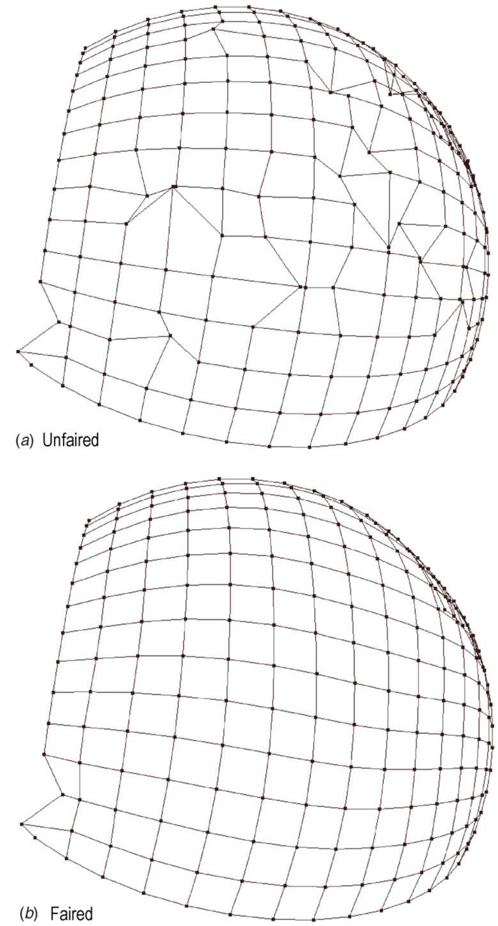


Fig. 1 15×15 Control net projected in E^3 for two-parameter bicubic B-spline spherical motion

5 Fine-Tuning of Two-Parameter Rational B-Spline Spherical Motions

We have seen earlier that the set of positions and orientations of an object that belong to a two-parameter rational B-spline spatial motion is given by Eq. (11). By equating the dual part of that equation to zero, we obtain the set of orientations that belong to a two-parameter rational B-spline spherical motion. By choosing the degree to (3;3) in Eq. (9), we obtain a bicubic rational B-spline image surface in the space of quaternions:

$$\mathbf{Q}^{n,m}(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} \mathbf{Q}_{i,j} N_{i,4}(u) N_{j,4}(v) \tag{19}$$

where the $N_{i,4}(u)$, $N_{j,4}(v)$ are the B-spline basis functions defined on the knot vectors:

$$\mathbf{u} = (u_i)_{i=0}^{n+4} \quad \mathbf{v} = (v_j)_{j=0}^{m+4}$$

and

$$u_i < u_{i+4} (i = 0, 1, \ldots, n) \quad v_j < v_{j+4} \quad (j = 0, 1, \ldots, m)$$

Quaternions $\mathbf{Q}_{i,j}$ are the B-spline image points that represent the given B-spline spherical displacements.

We now adapt the surface fairing algorithm (Hahmann [25]) to the space of quaternions to obtain an algorithm for the fine-tuning of two-parameter rational B-spline spherical motion.

Two-parameter spherical motion fairing algorithm:

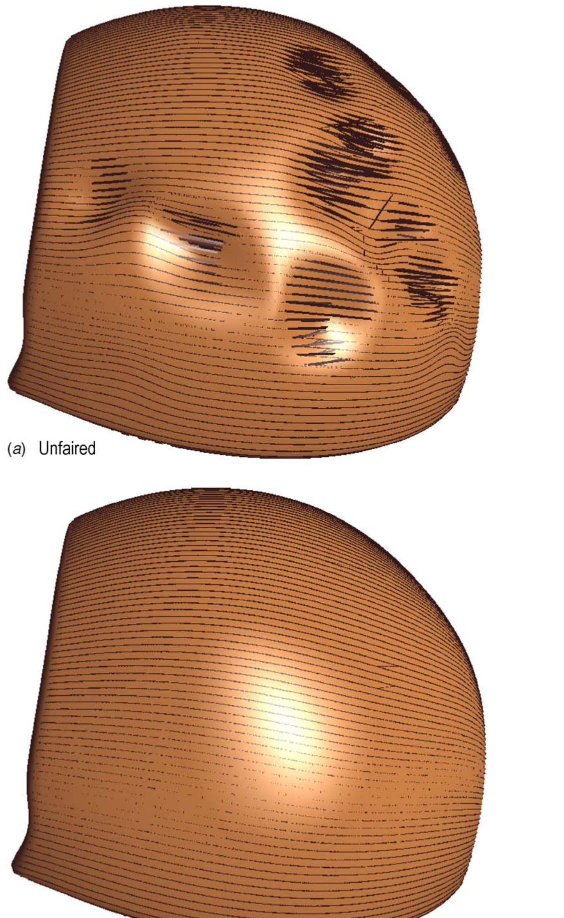
1. Convert the given rotation axis and rotation angle for each spherical displacement to quaternions ($\mathbf{Q}_{i,j}$) using Eq. (1). These quaternions represent the image points.
2. Calculate initial global fairness measure G_0 using Eq. (16).
3. DO
   - (a) FOR $i=4\ldots n$
     FOR $j=4\ldots m$
     - (i) Calculate $\Delta_{uuu}(u_i, v_j)$ and $\Delta_{vvv}(u_i, v_j)$ using Eq. (14)
     - (ii) Calculate and store L_{ij} using Eq. (15).
   - (b) Choose $(u_k; v_l)$ such that $L_{kl} = \max L_{ij}$.
   - (c) Apply local fairing step at $(u_k; v_l)$ using solution of Eqs. (18).
   - (d) Calculate new global measure G_k using Eq. (16).

WHILE $(|G_{k+1} - G_k| > \varepsilon)$, where ε is a very small predefined number.

4. Generate faired B-spline image surface, isophote, new control displacements, and motion trajectory.

Now we present an example that demonstrates this algorithm using 15×15 given spherical displacements. Figure 1 shows the control net structure obtained by projecting given image points ($\mathbf{Q}_{i,j}$; $i=0\ldots 14, j=0\ldots 14$) in E^3, while Fig. 2 shows the B-spline quaternion surface with the isophotes before and after the fine-tuning. It is apparent that both the control net and the quaternion surface appear regular (the wiggles and the dimples in the surface disappear) after the fine-tuning. The isophote lines are continuous

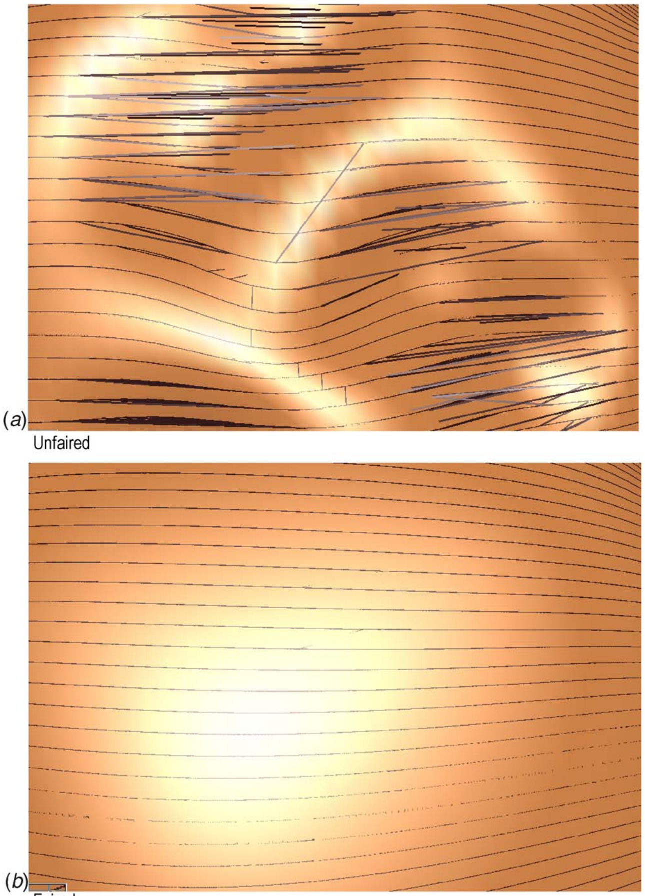
[1] In this paper, quaternion surfaces are visualized by projecting them in E^3.



(a) Unfaired

(b) Faired

Fig. 2 Quaternion surface with isophote for two-parameter bicubic B-spline spherical motion



(a) Unfaired

(b) Faired

Fig. 3 Isophote details on the quaternion surface for two-parameter bicubic B-spline spherical motion

indicating satisfactory smoothness (Fig. 3). Since the isophote lines are a function of the direction of light impacting on the surface, it is important to check the quality of the surface from different angles before concluding that surface is sufficiently smooth. It should be noted here that in ensuing discussion, we describe the surface and the motion in qualitative terms merely to corroborate the mathematical C^3 continuity achieved after the fairing process.

Next shown in Fig. 4 is the trajectory of "teapots," under a uniform bicubic B-spline spherical motion, both before and after fine-tuning. As is clear, after fine-tuning, the trajectory is better organized. Another indicator of better continuity in Fig. 4 is the orientation of the teapots around a single teapot. The variation of orientation in any small neighborhood of a single teapot is less extreme than in unfaired motion. The algorithm is fairly efficient since every local fairing step requires just a matrix to vector multiplication. The multiplied matrix (set up by the solution of Eq. (18)) is obtained by inversion only once in the beginning of the iterations. The order of complexity of the algorithm is dependent on the number of inner knots $((n-1)\times(m-1))$. In this example, we have chosen uniform parametrization for simplicity and faster execution but it is possible to choose centripetal, chord length, or other parametrizations that map the range behavior more appropriately to the domain (see Farin [28]). The execution time of the algorithm using uniform parametrization for this example was 5 s and it took 1930 iterations for the global fairness measure to reduce from $G_{\text{initial}}=7.84$ to $G_{\text{fair}}=0.002$ on a 2.6 GHz Pentium 4 system with 1 Gbyte RAM. Even though the algorithm was applied globally and it runs unattended, it allows restricting the motion fairing process to a small area of the initial B-spline motion by operating over a small subset of inner knots. It also allows to discard those steps of local fairing process, which change the motion beyond a user-defined acceptable level.

6 Fine-Tuning of Two-Parameter Rational B-Spline Spatial Motions

In the previous section, we discussed the fairing algorithm for two-parameter B-spline spherical motions. Since spatial displacements (as opposed to the spherical displacements) involve an extra translational term, the dual part of the given quaternions is nonzero. The only difference from the spherical case in fairing spatial motions is that, we must now consider both the real and dual parts of the quaternions.

A bicubic rational B-spline image surface is given by choosing the degree of basis functions to (3;3) in Eqs. (9)–(11):

$$\hat{\mathbf{Q}}^{n,m}(u,v)=\mathbf{Q}^{n,m}(u,v)+\varepsilon\mathbf{R}^{n,m}(u,v)$$

where $\mathbf{Q}^{n,m}(u,v)$ is given by Eq. (19), and

$$\mathbf{R}^{n,m}(u,v)=\sum_{i=0}^{n}\sum_{j=0}^{m}\mathbf{R}_{i,j}N_{i,4}(u)N_{j,4}(v)$$

The dual quaternion surface, $\hat{\mathbf{Q}}^{n,m}(u,v)$, represents a two-parameter rational B-spline motion of degree (6;6). The fairing process requires that both the real and the dual component be fine-tuned. The fine-tuning of the real part is already discussed in the previous section. In fine-tuning the dual part, there are two choices -one is to fine tune the dual part image surface given by

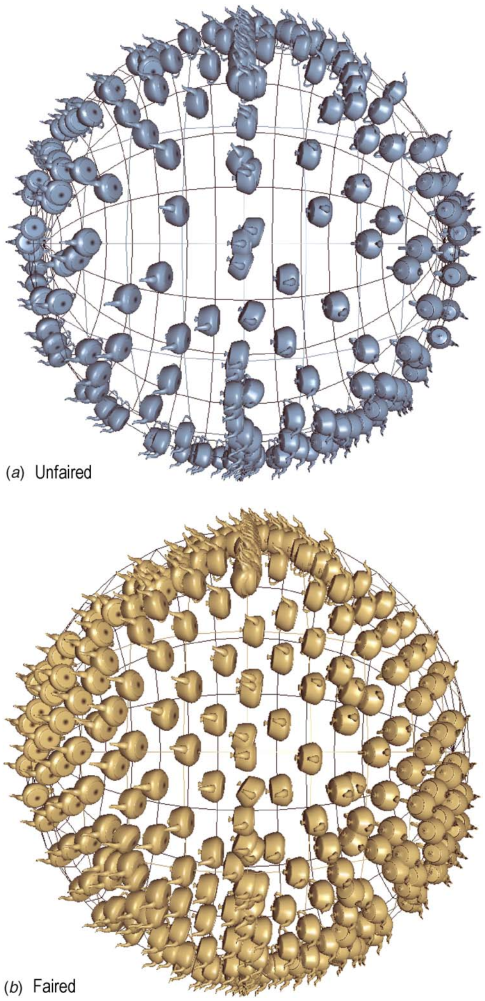


Fig. 4 Trajectory under two-parameter bicubic rational B-spline spherical motion

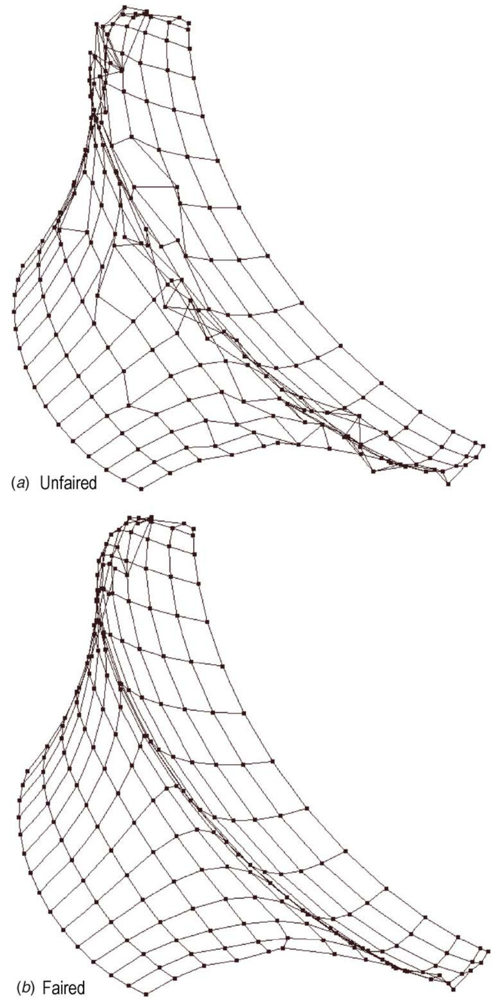


Fig. 5 15×15 Control net projected in E^3 for dual part under two-parameter bicubic B-spline spatial motion

Eq. (10) and the other is to fair the translation part directly. The surface defined by the translation vectors is called Translation surface. Here, we give an algorithm for fine tuning the real and the dual part of the dual quaternion surface. We note that this algorithm is only slightly different from the spherical version of it given in the previous section. The same algorithm can be applied for fairing the translation surface directly as well. They both yield fairer, though not identical surfaces.

Two-parameter spatial motion fairing algorithm:

1. Convert the given rotation axis, rotation angle, and translation for each spatial displacement to dual quaternions ($\mathbf{Q}_{i,j} + \varepsilon \mathbf{R}_{i,j}$) using Eqs. (1) and (7). These dual quaternions represent the image points of the dual projective space.
2. Calculate initial global measure G_0 for both the real and the dual part using Eq. (16).
3. DO for both the real and the dual part
   (a) FOR $i=4\ldots n$
   FOR $j=4\ldots m$
      (i) Calculate $\Delta_{uuu}(u_i,v_j)$ and $\Delta_{vvv}(u_i,v_j)$ using Eq. (14).
      (ii) Calculate and store L_{ij} using Eq. (15).
   (b) Choose $(u_k;v_l)$ such that $L_{kl}=\max L_{ij}$.
   (c) Apply local fairing step at $(u_k;v_l)$ using solution of Eqs. (17) and (18).
   (d) Calculate new global measure G_k using Eq. (16).

WHILE ($|G_{k+1}-G_k|>\varepsilon$), where ε is a very small predefined number.

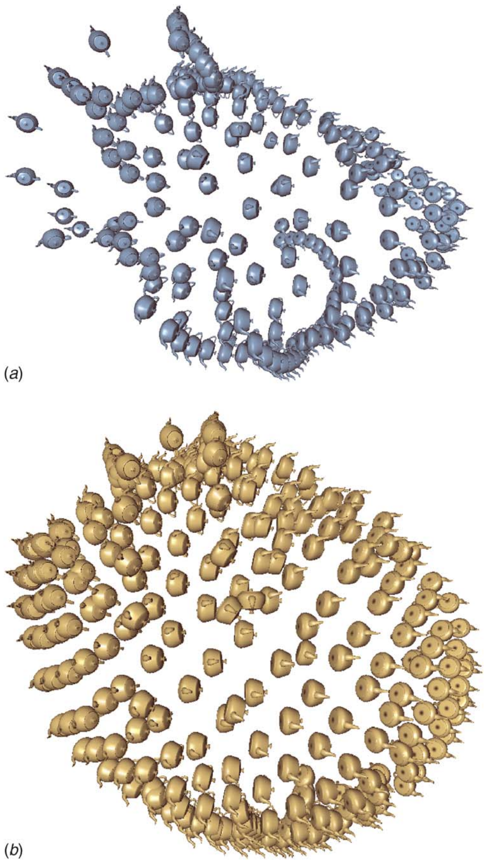
4. Generate faired B-spline image surface, isophote, new control displacements, and motion trajectory.

Now, we illustrate the implementation of spatial fine-tuning algorithm by an example. Fine-tuning of the real part is already illustrated in Sec. (5) and we use the same value for the real part in the succeeding examples. Here, we only illustrate fine-tuning of the dual part. Figure 5 shows the control net structure for the dual part before and after fairing. The control net is much more regular and well behaved after the fairing process.

Figure 6 shows the trajectories of teapots before and after fine-tuning, respectively. The effect of fine tuning is much more pronounced in this case. Before fairing the dual quaternion image surface, motion seems chaotic even though it is a C^2 continuous motion. After fairing, the order in the motion and the control displacements improve. On the same hardware system, the runtime of the algorithm was 8 s (3052 iterations) and it reduced the global fairness measure from $G_{\text{initial}}=10.195$ to $G_{\text{fair}}=0.003$.

7 Conclusions

A method is presented for automatic fairing of two-parameter rational B-spline motion by extending the method for surface fair-



(*a*)

(*b*)

Fig. 6 Trajectory under two-parameter bicubic rational B-spline spatial motion

ing to the space of dual quaternions. The resulting algorithm is efficient, automatic, local in approach, and increases the continuity of the motion to C^3. By restricting the number of control points involved at every step of fairing or reducing the number of inner knot points, one can further localize the effect of fairing. The automatic fairing algorithm may be used to fine-tune two-parameter B-spline tool motion for five-axis NC machining process as well as for motion specifications for robotic and virtual reality systems. By making both parameters a function of time, one can obtain a one-parameter motion that inherits the smoothness of the faired two-parameter motion.

Acknowledgment

This work was supported by National Science Foundation Grant Nos. DMI-9800690 and DMI-0500064 to Stony Brook University.

References

[1] Bottema, O., and Roth, B., 1979, *Theoretical Kinematics*, North Holland, Amsterdam.

[2] Shoemake, K., 1985, "Animating Rotation With Quaternion Curves," *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques*, ACM, New York, NY, pp. 245–254.

[3] Nielson, G. M., and Heiland, R. W., 1992, "Animated Rotations Using Quaternion and Splines on a 4D Sphere," *Programmirovanie* (Russia), Springer-Verlag, pp. 17–27; English edition, *Programming and Computer Software*, Plenum, New York, Vol. 18, pp. 145–154.

[4] Kim, M. S., and Nam, K. W., 1995, "Interpolating Solid Orientations With Circular Blending Quaternion Curves," Comput.-Aided Des., **27**(5), pp. 385–398.

[5] Kim, M.-J., Kim, M.-S., and Shin, S. Y., 1995, "A C^2 Continuous B-Spline Quaternion Curve Interpolating a Given Sequence of Solid Orientations," *Proceedings of the Computer Animation, Proceedings of the Computer Animation*, IEEE Computer Society, Washington, DC, pp. 19–21.

[6] Ge, Q. J., and Ravani, B., 1994, "Geometric Construction of Bezier Motions," ASME J. Mech. Des., **116**(3), pp. 749–755.

[7] Ge, Q. J., and Ravani, B., 1994, "Computer-Aided Geometric Design of Motion Interpolants," ASME J. Mech. Des., **116**(3), pp. 756–762.

[8] Purwar, A., and Ge, Q. J., 2005, "On the Effect of Dual Weights in Computer Aided Design of Rational Motions," ASME J. Mech. Des., **127**(5), pp. 967–972.

[9] Juttler, B., and Wagner, M. G., 1996, "Computer-Aided Design with Spatial Rational B-Spline Motions," ASME J. Mech. Des., **118**(2), pp. 193–201.

[10] Röschel, O., 1998, "Rational Motion Design—A Survey," Comput.-Aided Des., **30**(3), pp. 169–178.

[11] Purwar, A., 2005, "Design and Fine Tuning of One- and Two-Parameter Rational Motion," Ph.D. thesis, State University of New York at Stony Brook, Stony Brook, NY.

[12] Srinivasan, L. N., and Ge, Q. J., 1998, "Fine Tuning of Rational B-Spline Motions," ASME J. Mech. Des., **120**(1), pp. 46–51.

[13] Fang, Y. C., Hsieh, C. C., Kim, M. J., Chang, J. J., and Woo, T. C., 1998, "Real Time Motion Fairing With Unit Quaternions," Comput.-Aided Des., **30**(3), pp. 191–198.

[14] Lee, J., and Shin, S., 1996, "Motion Fairing," *Proceedings Computer Animation '96*, pp. 136–143.

[15] Kim, M. J., Hsieh, C. C., Wang, M. E., Wang, C. K., Fang, Y. C., and Woo, T. C., 1996, "Noise Smoothing for VR Equipment in the Quaternion Space," *Proceedings of the Virtual Reality in Manufacturing Research and Education*.

[16] Lee, J., and Shin, S., 2002, "General Construction of Time-Domain Filters for Orientation Data," IEEE Trans. Vis. Comput. Graph., **8**(2), pp. 119–128.

[17] Hsieh, C. C., 2002, "Motion Smoothing Using Wavelets," J. Manuf. Sci. Eng., **35**(2), pp. 157–169.

[18] Hsieh, C. C., 2001, "B-Spline Wavelet-Based Motion Smoothing," Comput. Ind. Eng., **41**(1), pp. 59–76.

[19] Hsieh, C. C., and Chang, T. Y., 2003, "Motion Fairing Using Genetic Algorithms," Comput.-Aided Des., **35**(8), pp. 739–749.

[20] Ge, Q. J., and Sirchia, M., 1999, "Computer Aided Geometric Design of Two-Parameter Freeform Motions," ASME J. Mech. Des., **121**(4), pp. 502–506.

[21] Ge, Q. J., 1996, "Kinematics-Driven Geometric Modeling: A Framework for Simultaneous Tool-Path Generation and Sculptured Surface Design," *IEEE Robotics and Automation Conference*, Minneapolis, MN, Vol. 2, pp. 1819–1824.

[22] Kjellander, J. A., 1983, "Smoothing of Bicubic Parametric Surfaces," Comput.-Aided Des., **15**(5), pp. 289–293.

[23] Nowacki, H., and Reese, D., 1983, "Design and Fairing Ship Surfaces," *Surfaces in CAGD*, R. Barnhill, and W. Bohm, eds., North-Holland, Amsterdam, pp. 121–134.

[24] Lott, N. J., and Pullin, D. I., 1988, "Method for Fairing B-Spline Surfaces," Comput.-Aided Des., **20**(10), pp. 597–604.

[25] Hahmann, S., 1998, "Shape Improvement of Surfaces," *Geometric Modelling, Computing* [Suppl.], G. Farin, H. Bieri, G. Brunett, and T. DeRose, eds., Vol. 13, Springer-Verlag, pp. 135–152.

[26] McCarthy, J. M., 1990, *Introduction to Theoretical Kinematics*, MIT, Cambridge, MA.

[27] Ravani, B., and Roth, B., 1984, "Mappings of Spatial Kinematics," ASME J. Mech., Transm., Autom. Des., **106**(3), pp. 341–347.

[28] Farin, G., 1996, *Curves and Surfaces for Computer-Aided Geometric Design: A Practical Guide*, 4th ed. Academic, New York.

[29] Piegl, L., and Tiller, W., 1995, *The Nurbs Book*, Springer, Berlin.

[30] Schumaker, L. L., 1981, *Spline Functions: Basic Theory*, Wiley, New York.

[31] De Boor, C., 1978, *A Practical Guide to Splines*, Springer, New York.

[32] Welch, W., and Witkin, A., 1992, "Variational Surface Modeling," *SIGGRAPH '92*, Vol. 26, Series 2, pp. 157–166.

[33] Wesselink, W., and Veltcamp, R. C., 1995, "Interactive Design of Contrained Variational Curves," Comput. Aided Geom. Des., **12**(5), pp. 533–546.

[34] Greiner, G., and Seidel, H. P., 1997, "Automatic Modeling of Smooth Spline Surfaces," *Winter School of Computer Graphics'97*, N. Magnenat-Thalmann and V. Skala, eds., pp. 665–675.

[35] Hoschek, J., and Lasser, D., 1993, *Fundamentals of Computer Aided Geometric Design*, A. K. Peters, Wellesley, MA.

[36] Poeschl, T., 1984, "Detecting Surface Irregularities Using Isophotes," Comput. Aided Geom. Des., **1**(2), pp. 163–168.