Proceedings of IDETC/CIE 2005
ASME 2005 International Design Engineering Technical Conferences
& Computers and Information in Engineering Conference
September 24-28, 2005, Long Beach, California USA

DETC2005-85458

AUTOMATIC FAIRING OF TWO-PARAMETER RATIONAL B-SPLINE MOTION

Anurag Purwar, Xiaoyi Chi, Qiaode Jeffrey Ge

Department of Mechanical Engineering
State University of New York
Stony Brook, New York 11794-2300
purwar@design.eng.sunysb.edu

ABSTRACT

This paper deals with the problem of automatic fairing of two-parameter B-Spline spherical and spatial motions. The concept of two-parameter freeform motions brings together the notion of the analytically determined two-parameter motions in Theoretical Kinematics and the concept of freeform surfaces in the field of Computer Aided Geometric Design (CAGD). A dual quaternion representation of spatial displacements is used and the problem of fairing two-parameter motions is studied as a surface fairing problem in the space of dual quaternions. By combining the latest results in surface fairing from the field of CAGD and computer aided synthesis of freeform rational motions, smoother (C^3 continuous) two-parameter rational B-Spline motions are generated. The results presented in this paper are extensions of previous results on fine-tuning of one-parameter B-spline motions. The problem of motion smoothing has important applications in the Cartesian motion planning, camera motion synthesis, spatial navigation in visualization, and virtual reality systems. Several examples are presented to illustrate the effectiveness of the proposed method.

1 INTRODUCTION

Devising a smooth motion is an important problem in cartesian motion planning, camera motion synthesis, spatial navigation in visualization, animation, and virtual reality systems. Generally in these applications, it is harder to construct an ab initio motion free of undesired discontinuities. To overcome this problem, six DOF rigid body samples of orientation and translation are taken from a real world motion and then these samples are interpolated or approximated to synthesize a motion. However, due to sensing errors incurred during the sampling process, an acceptable reconstruction of the motion is hard to achieve. There are also situations when even if originally motion can be synthesized using keyframe editors, the constructed motion possess jerkiness and the speed variations are unacceptable. This necessitates that the constructed motion be smoothed (or, faired), either globally or locally, within specified constraints.

Smoothing of a rigid body motion consists of smoothing two components of spatial displacement – rotation and translation. However, smoothing of the two components is dependent on their mathematical representation. Translation, as an element of vector space, can be faired using well-known curve-fairing techniques from CAGD (Farin [1], Farin et al. [2], Sapidis and Farin [3], Kjellander [4]). Rotation as an element of SO(3) have had representation issues – using Euler angles gives rise to the Gimbal lock problem (Kane [5]) while in the matrix form orthogonality is hard to preserve during the interpolation process (Fillmore [6], Roschel [7]). Quaternion representation of rotation (Bottema and Roth [8]) is widely recognized to be an effective way of dealing with the aforementioned problems (Shoemake [9], Nielson and Heiland [10], Nielson [11], Kim and Nam [12], Kim et al. [13]).

A much more elegant representation of spatial displacements is obtained by combining translation and rotation using Dual Quaternion (McCarthy [14], Ge and Ravani [15]), which is a slightly modified version of Study's Soma parameters (Bottema and Roth [8], Study [16]). It is this representation that we use in this paper. Juttler [17], Juttler and Wagner [18], and Wagner [19, 20] also used a quaternion based representation by essentially treating translation and rotation separately.

There has been a great deal of research on the problem of motion synthesis, which is defined as the problem of designing an approximation or interpolation to a given set of 6 DOF displacements (translation and orientation). Ge and Ravani [15, 21, 22] developed a new framework for geometric constructions of spatial motions by combining the concepts from Kinematics and CAGD. Purwar and Ge [24] used dual numbers as the ambient scalars of such spatial motions to wrest finer control over the synthesis of such motions. Juttler [17], Juttler and Wagner [18], and Wagner [19,20] provided a rich set of tools and algorithms for motion design.

Despite these advances in the motion synthesis field, there has been much less research on the motion fairing problem. Srinivas and Ge [25] fine tuned rational B-spline motions by fine-tuning rational curves in its ambient projective dual three-space. They provided algorithms for both path and speed smoothing, which automatically removed the third order discontinuities in the path of a cubic B-spline curve in the Image Space and also obtained a near constant kinetic energy paramterization for the speed smoothing algorithm. In the intervening years, a few researchers have used various techniques to smooth out the motion – Fang et al [26] used a lowpass filter coupled with an adaptive, meadiative filter for angular velocities to achieve smooth rotation; Lee and Shin [27] used an algorithm that iteratively minimizes the energy function reflecting the forces and torques, exerted on a moving object; Kim et al. [28] 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 non-linear nature of orientation space, they are not directly applicable. Lee and Shin [29] 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, Hsieh [30, 31] used wavelets based approach and Hsieh and Chang [32] used genetic algorithms to do motion fairing.

The idea of motion synthesis and fairing discussed so far concerns itself with the usual sense of Kinematics' definition of one-parameter motion, in which the position of a moving object depends on a single parameter, most often identified with the time t. The trajectory generated by a moving point, line, or plane is a curve, ruled surface, or a developable surface respectively. Bottema and Roth [8] studied the kinematics of analytically defined n-parameter motions. Kinematics of multi-degree of freedom motions constrained by mechanical joints have been investigated in the field of Robotics (see Angeles [33] and Gupta [34]). For a two-parameter motion, the locus of a point is, in general, a surface, called its trajectory surface; that of a line is its trajectory congruence; and that of a plane is the set of tangent planes of a surface, enveloped by the plane. Ge and Sirchia [35] studied the problem of computer aided geometric design of two-parameter freeform motions by combining CAGD methods with the kinematics of two-parameter motion. One of the motivations for studying two-parameter freeform motions is to develop a kinematics based approach to geometric shape design and 5-axis NC tool path planning (Ge [36], Zhang et al. [37]).

If the problem of one-parameter motion fairing maps naturally to the fine tuning of rational curve in projective dual three-space, the problem of two-parameter motion fairing would map to fine tuning of rational surface in projective space (Image Space). In this paper, for the first time, we bring together kinematic geometry of two-parameter B-Spline motions, computer aided geometric design, and the fairing of surfaces (Kjellander [38], Nowacki [39], Farin [1], Lott and Pullin [40]) to develop an innovative method for fine tuning of two-parameter B-Spline motions. We use an improved surface fairing technique proposed by Hahmann [41] to do the fine tuning of the Dual Quaternion Surface. In doing so, the algorithm automatically detects and removes third order discontinuities in the Image Space and elevate the continuity of the B-Spline motion from C^2 to C^3.

The rest of the paper is organized as follows. Section 2 discusses a few representation of spatial displacements with an emphasis on dual quaternion based representation that we use in this paper. Section 3 presents the notion of two-parameter rational B-Spline motion and discusses some of its properties. Section 4 presents a general overview of fine tuning two parameter rational motion, while also giving a local fairness criterion and a surface irregularity visualization technique called isophote. Section 5 and Section 6 present the algorithm for fine tuning of two parameter rational B-Spine spherical and spatial motion respectively along with a few visualizations before concluding the paper.

2 REPRESENTATION OF SPATIAL DISPLACEMENTS

A spatial displacement of a rigid body is commonly represented by the following transformation of a moving frame M attached to the moving body with respect to a fixed frame F attached to the fixed space:

$$\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \begin{bmatrix} [R] & |\mathbf{d} \\ \hline 000| & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}. \quad (1)$$

where $[R]$ is an orthogonal matrix representing a rotation and d is a vector representing a translation; $\mathbf{X}$ and $\mathbf{x}$ are vectors whose scalar components are the Cartesian coordinates of the point as measured in F and M, respectively. The use of such matrix representation, however, is not convenient when dealing with the problem of synthesizing a rational motion that interpolates or approximates a set of displacements. One of the main obstacles is to the issue of preserving the orthogonality of the rotation matrix in the interpolation/approximation process (Fillmore [6], Roschel [7]). In this paper, we use well established dual quaternion representation(Ge and Ravani [15]) for spatial displacements. In what follows, we review the concepts of quaternions and dual quaternions in so far as necessary for the development of the current paper.

2.1 Unit Quaternion as Rotation

For any rotation in Cartesian space, there exists an equivalent rotation axis and the rotation angle about this axis. Let $\mathbf{s} = (s_x, s_y, s_z)$ denote a unit vector along the axis of the rotation and θ denote the angle of rotation. The axis $\mathbf{s}$ and angle θ of the rotation can be used to define the so-called *Euler-Rodrigues parameters*.

$$q_1 = s_x \sin(\theta/2),\ q_2 = s_y \sin(\theta/2),\ q_3 = s_z \sin(\theta/2),\ q_4 = \cos(\theta/2)$$

The Euler parameters and the quaternion units, $1, i, j, k$ can be combined to define a quaternion of rotation:

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

For more details on quaternion algebra, see Bottema and Roth [8] or McCarthy [14]. Let $\mathbf{x} = (x_1, x_2, x_3)$, and $\mathbf{X} = (X_1, X_2, X_3)$ denote Cartesian coordinates of a point before and after a rotation respectively. They can be represented as vector quaternions:

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

The coordinate transformation from $\mathbf{x}$ to $\mathbf{X}$ is represented by the quaternion product:

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

where $\mathbf{Q}^{-1}$ denotes the inverse of $\mathbf{Q}$.

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

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

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

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

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

where the rotation matrix $[R]$ is given by

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

It should be noted here that $\mathbf{Q}$ can also be treated as homogeneous coordinates of a rotation since multiplying each one of the *Euler-Rodrigues parameters* by a scalar leaves the rotation matrix $[R]$ invariant.

2.2 Dual Quaternion as Spatial Displacement

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

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

where $\mathbf{d} = d_1 i + d_2 j + d_3 k$ is a vector quaternion corresponding to the translation component, which is defined by the non-homogeneous parameters (d_1, d_2, d_3). The translation component can be alternatively represented using homogeneous coordinates $\mathbf{D} = (D_1, D_2, D_3, D_4)$, where

$$D_1 = kd_1,\ D_2 = kd_2,\ D_3 = kd_3,\ D_4 = k.$$

This homogeneous representation of spatial displacements was discussed by Ravani and Roth [23] in the context of kinematics mappings. The set of eight homogeneous parameters $\mathbf{Q}$, $\mathbf{D}$ has been used by Jüttler [17], Wagner [19, 20], and Jüttler and Wagner [18], for computer-aided design of rational motions. While this formulation allows direct application of the existing CAGD techniques to motion design, the resulting motions are not completely reference-frame invariant and depend on the choice of the origins of the reference frames. In this paper, we do not use this formulation but follow McCarthy [14] and Ge and Ravani [15] and use a slightly modified version of Study's Soma parameters to represent the spatial displacements.

Study's parameters are given by another set of eight homogeneous parameters ($\mathbf{Q}$, $\mathbf{Q}^0$) where $\mathbf{Q} = (Q_1, Q_2, Q_3, Q_4)$ represents the quaternion of homogeneous Euler parameters of rotation and $\mathbf{Q}^0 = (Q_1^0, Q_2^0, Q_3^0, Q_4^0)$ is another quaternion whose components are given by

$$\begin{bmatrix} Q_1^0 \\ Q_2^0 \\ Q_3^0 \\ Q_4^0 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 0 & -d_3 & d_2 & d_1 \\ d_3 & 0 & -d_1 & d_2 \\ -d_2 & d_1 & 0 & d_3 \\ -d_1 & -d_2 & -d_3 & 0 \end{bmatrix}\begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \end{bmatrix}. \tag{5}$$

The translation vector $\mathbf{d}$ (d_1, d_2, d_3) can be recovered from Eq.(5) in terms of $(\mathbf{Q}, \mathbf{Q}^0)$ by using following

$$\text{d} = -\frac{2}{S^2}\begin{bmatrix} Q_4^0 Q_1 - Q_1^0 Q_4 + Q_2^0 Q_3 - Q_3^0 Q_2 \\ Q_4^0 Q_2 - Q_2^0 Q_4 + Q_3^0 Q_1 - Q_1^0 Q_3 \\ Q_4^0 Q_3 - Q_3^0 Q_4 + Q_1^0 Q_2 - Q_2^0 Q_1 \end{bmatrix}. \tag{6}$$

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

Study's parameters can also be written in dual quaternion form as

$$\hat{\mathbf{Q}} = \mathbf{Q} + \varepsilon\mathbf{Q^0} \tag{7}$$

where ε is the dual unit with the property $\varepsilon^2 = 0$ (see Bottema and Roth [8] for details on dual number). In quaternion form, Eqs. (5) and (6) can be written more concisely as follows, respectively:

$$\mathbf{Q}^0 = (1/2)\mathbf{dQ}, \tag{8}$$

$$\mathbf{d} = \frac{(\mathbf{Q}^0)\mathbf{Q}^* - \mathbf{Q}(\mathbf{Q}^0)^*}{\mathbf{QQ}^*}, \tag{9}$$

where $\mathbf{d}$ is a vector quaternion, which has no scalar part, and $\text{Q}^* = (-Q_1, -Q_2, -Q_3, Q_4)$ is the conjugate of $\mathbf{Q}$ such that $\mathbf{QQ}^* = Q_1^2 + Q_2^2 + Q_3^2 + Q_4^2$. Note that Eq. (6) or Eq. (9) can be used to recover $\mathbf{d}$ from $\mathbf{Q}$ and $\mathbf{Q}^0$ even when they do not satisfy the well-known Plücker condition:

$$Q_1 Q_1^0 + Q_2 Q_2^0 + Q_3 Q_3^0 + Q_4 Q_4^0 = 0. \tag{10}$$

However, when the dual quaternion components satisfy the above Plücker condition, Eq. (9) reduces to the following well-known equation

$$\mathbf{d} = \frac{2(\mathbf{Q}^0)\mathbf{Q}^*}{\mathbf{QQ}^*}$$

which follows directly from Eq. (8).

It is instructive to note here that $(\mathbf{Q}, \mathbf{Q}^0)$ serve as homogeneous coordinates of spatial displacements since multiplying them by a non-zero scalar yields the same rotation matrix and translation vector $\mathbf{d}$. Ravani and Roth [23] considered $\hat{\mathbf{Q}} = (\hat{Q}_1, \hat{Q}_2, \hat{Q}_3, \hat{Q}_4)$ as a set of four homogeneous dual coordinates that define a point in a projective dual three-space, called the Image Space of spatial displacements. A curve in this projective space corresponds to a one-parameter motion in Cartesian space while a surface corresponds to a two-parameter motion. We develop this notion further in the next section.

3 TWO-PARAMETER RATIONAL B-SPLINE MOTION

Given an $(n+1)\times(m+1)$ array of dual quaternions $\hat{\mathbf{Q}}_{i,j}$, we may define the following tensor-product B-Spline surface in the space of dual quaternions:

$$\mathbf{Q}^{n,m}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m}\mathbf{Q}_{i,j}N_i^n(u)N_j^m(v) \tag{11}$$

$$\mathbf{R}^{n,m}(u,v) = \sum_{p=0}^{n}\sum_{q=0}^{m}\mathbf{R}_{p,q}N_p^n(u)N_q^m(v) \tag{12}$$

where $\mathbf{Q}_{i,j}$ is the real part of the dual quaternion $\hat{\mathbf{Q}}^{n,m}(u,v)$, $\mathbf{R}_{p,q}$ is the dual part of the dual quaternion $\hat{\mathbf{Q}}^{n,m}(u,v)$. $N_i^n(u)$,$N_j^m(v)$,$N_p^n(u)$ and $N_q^m(v)$ are B-Spline basis functions. The dual quaternion surface,

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

represents the set of positions and orientations of an object that belong to a two parameter B-Spline motion.

Two parameter B-Spline motions enjoy some properties as follows:

Coordinate-frame invariance The representations of two parameter rational B-Spline motions in terms of dual quaternions are invariant with respect to change of both the fixed and the moving reference frames.

Convex hull property Like the B-Spline surfaces, the two parameter motions have the local convex hull property.

Bounded When $u, v = 0, 1$, we obtain four one parameter motions, which may be referred to as *boundary motions*. Their B-Spline control positions are given by the boundary control positions of the array (or net) of B-Spline control positions. In particular, the four corner positions of the B-Spline control net belong to the two parameter motion.

Local control Moving a B-Spline control position changes the motion locally only. This enables fine tuning of such motions locally to get smoother motions.

A bi-variate equation of motion of a point in Cartesian space can be obtained by recasting Eq. (1) in terms of dual quaternions and subsequently substituting for the expressions of $\mathbf{Q}^{n,m}(u,v)$ and $\mathbf{R}^{n,m}(u,v)$ from Eq. (11) and Eq. (12).

$$\tilde{\mathbf{P}}(u,v) = \mathbf{Q}(u,v)\mathbf{P}\mathbf{Q}(u,v)^* + p_4[\mathbf{R}(u,v)\mathbf{Q}(u,v)^* - \mathbf{Q}(u,v)\mathbf{R}(u,v)^*] \tag{13}$$

where $\mathbf{Q}(u,v)^*$and $\mathbf{R}(u,v)^*$ are conjugate of $\mathbf{Q}(u,v)$ and $\mathbf{R}(u,v)$ respectively and $\tilde{\mathbf{P}}$ denotes homogeneous coordinates of a point $\mathbf{P}$:(p_1, p_2, p_3, p_4) of the object after the displacement. Equation (13) defines trajectory surface of the point $\mathbf{P}$. A derivation of the Eq. (13) is available in the Appendix of Purwar and Ge [24].

4 FINE TUNING TWO-PARAMETER RATIONAL B-SPLINE MOTION

In the earlier sections, using the methods of CAGD, we established the synthesis of a two-parameter B-Spline motion as the construction of a surface in the projective space. Thus, the fine tuning of a two-parameter motion can be construed as a surface fairing step in projective space. In CAGD, the surface is modeled as patches connected along their boundaries. The continuity of the patches at the boundaries determines the smoothness of the surface. Continuity, characterized as the rth order derivative continuity (C^r), in turn depends on the order of interpolation and the type of control desired.

Surfaces can be faired using *fairness constraints* or by performing *post-processing fairing*. The former method usually incorporates a linearized physical based fairness criterion in the interpolation or approximation. These methods which are usually non-linear in nature try to create fair shapes by minimizing the energy of a curve or surface under given constraints. They have been introduced for surfaces by Welch and Witkin [42]. Wesselink and Veltcamp [43] present variational modeling techniques and tools for curves and surfaces. Greiner and Seidel [44] 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 can not be achieved.

The second method tries to remove unwanted surface wiggles by smoothing the control net, which defines the surface(Kjellander [38]). It is a variation of this method that we employ in this paper. Our method is based on Hahmann [41]'s *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. Hahmann's algorithm is also flexible since it allows for different fairing steps, permits localised fairing process, and can impose tolerance control.

Fine Tuning Criteria

Given an $(n+1)\times(m+1)$ array of control points, a bicubic B-Spline surface can be defined as:

$$\mathbf{X}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} \mathbf{d}_{i,j} N_{i,4}(u) N_{j,4}(v), \tag{14}$$

The knot sequences are as follows:

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

and

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

Hahmann [41] defines a fairing criterion as follows: "A B-Spline surface $\mathbf{X}(u,v)$ of class C^2 is fairer at the knot $[u_k, v_l]$, $(k,l) \in I$, if $\mathbf{X}$ is C^3 at $[u_k, v_l]$."

Bicubic B-Spline surfaces have all mixed third order partial derivatives (i.e. $\mathbf{X}_{u^\nu v^\mu}$ with $\nu, \mu \geq 1$) 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 difference of $\mathbf{X}_{uuu}$ in u-direction and $\mathbf{X}_{vvv}$ in v-direction at the knot $[u_k, v_l]$ is defined to be the local fairness measure.

Discontinuity vectors (Hahmann [41]) are given by:

$$\begin{aligned}
&\Delta_{uuu}(u_k,v_l) = \mathbf{X}_{uuu}(u_k^-,v_l) - \mathbf{X}_{uuu}(u_k^+,v_l) \\
&= \textstyle\sum_{j=0}^{m} \mathbf{d}_{k-1,j}^{(3,0)} N_{j,4}(v_l) - \sum_{j=0}^{m} \mathbf{d}_{k,j}^{(3,0)} N_{j,4}(v_l) \\
&= \textstyle\sum_{i=k-4}^{k}\sum_{j=l-3}^{l-1} \alpha_{ij}\mathbf{d}_{ij} \\
&\Delta_{vvv}(u_k,v_l) = \mathbf{X}_{vvv}(u_k,v_l^-) - \mathbf{X}_{vvv}(u_k,v_l^+) \\
&= \textstyle\sum_{i=0}^{n} \mathbf{d}_{i,l-1}^{(0,3)} N_{i,4}(u_k) - \sum_{i=0}^{n} \mathbf{d}_{i,l}^{(0,3)} N_{i,4}(u_k) \\
&= \textstyle\sum_{i=k-3}^{k-1}\sum_{j=l-4}^{l} \beta_{ij}\mathbf{d}_{ij}
\end{aligned} \tag{15}$$

with

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

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

$$L_{kl} = \|\Delta_{uuu}(u_k, v_l)\|^2 + \|\Delta_{vvv}(u_k, v_l)\|^2 \tag{16}$$

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

$$G_X = \sum_{(k,l)\in I} L_{kl} \tag{17}$$

Hahmann [41] uses a classic least-squares approximation with constraints to solve this problem

$$Minimize\ F(\hat{\mathbf{d}}_{ij}) = \sum_{i=k-3}^{k-1} \sum_{j=l-3}^{l-1} \left\|\mathbf{d}_{ij} - \hat{\mathbf{d}}_{ij}\right\|^2 \tag{18}$$

subject to $\Delta_{vvv} = 0,\ \Delta_{uuu} = 0$

where $\mathbf{d}_{ij}$ and $\hat{\mathbf{d}}_{ij}$ denote control points before and after an update. Even though *discontinuity vector* involves more than nine control points, Eq. (18) indicates that only nine innermost points are chosen for an update at every step of fairing.

The Lagrange multipliers method is used to solve this problem:

$$\Phi(\hat{\mathbf{d}}_{ij}, \lambda, \mu) = F(\hat{\mathbf{d}}_{ij}) + \lambda(\Delta_{uuu}(u_k, v_l)) + \mu(\Delta_{vvv}(u_k, v_l)) \rightarrow min \tag{19}$$

Use of Isophote

In this paper, the Isophote technique (Poeshcl [45]) is used for obtaining a visual feedback on the fairness of the surface imperfections. Connecting all points of a surface along which the angle α between the light direction $\mathbf{L}$ and the surface normal $\mathbf{N}$ is constant and in the range between -90 and 90 degrees we obtain a line of constant illumination intensity called isophote. Only curves with angles between 0 and 90 degrees are in the illumination part.

Let $x(u, v)$ be the parametric representation of a surface of C^2 continuity and

$$\mathbf{N}(u, v) = \frac{\mathbf{x}_u \times \mathbf{x}_v}{|\mathbf{x}_u \times \mathbf{x}_v|}$$

the unit surface normal. Then for each point $x(u, v(u))$ of an isophote of illumination with light direction $\mathbf{L}$, the relation

$$\mathbf{N}(u, v) \cdot \mathbf{L} = \cos(\alpha) = C \tag{20}$$

holds with a constant $C \in [-1, +1]$ and $\alpha \in [-90^\circ, +90^\circ]$. By differentiation it follows that

$$(\mathbf{N}_u \cdot \mathbf{L})du + (\mathbf{N}_v \cdot \mathbf{L})dv = 0$$

or for $(\mathbf{N}_v \cdot \mathbf{L}) \neq 0$:

$$\frac{dv}{du} = -\frac{\mathbf{N}_u \cdot \mathbf{L}}{\mathbf{N}_v \cdot \mathbf{L}} \tag{21}$$

The surface continuity and the isophote continuity have the following relationship: For a surface with C^r continuity, the corresponding isophote is C^{r-1} continuous.

5 FINE TUNING OF TWO-PARAMETER RATIONAL B-SPLINE SPHERICAL MOTIONS

A bicubic rational B-Spline image surface is given by:

$$\hat{\mathbf{X}}(u, v) = \sum_{i=0}^{n} \sum_{j=0}^{m} \hat{\mathbf{d}}_{i,j} N_{i,4}(u) N_{j,4}(v), \tag{22}$$

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

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

and $\hat{\mathbf{d}}_{i,j}$ are the B-Spline image points that represent the B-Spline control displacements. Note in the case of spherical motions, we only consider the real part for the dual quaternions. Let $\hat{\mathbf{X}} = \mathbf{X} + \varepsilon\mathbf{X}^0$, $\hat{\mathbf{d}}_{i,j} = \mathbf{d}_{i,j} + \varepsilon\mathbf{d}^0_{i,j}$. For spherical motions, the dual parts are set to be zero, $\mathbf{X}^0 = 0, \mathbf{d}^0_{i,j} = 0$.

Based on the surface fine tuning algorithm (Hahmann [41]) discussed earlier, we give the algorithm for fine tuning two parameter rational B-Spline spherical and spatial motion in table 1.

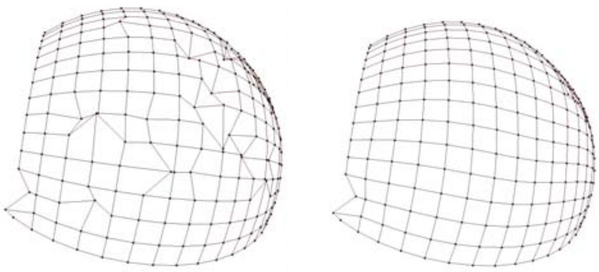
Now, we present a few visualizations using the aforementioned fine tuning algorithm. Figure 1 shows the control net

Table 1. Algorithm for fine tuning two parameter rational B-Spline spherical and spatial motion

(1) Input $n \times m$ control data which includes rotation axis, rotation angle, translation and knot vector.
(2) Convert the control rotation axis, rotation angles, translations to dual quaternions which represent the control displacements. For spherical motion, set the dual part of the quaternions to zero.
(3) Calculate initial global measure G^0: Perform steps (a), (b), and (c) for both the dual and real part in case of spatial motion and only for the real part in case of spherical motion.
(a) In the image space, calculate $\Delta_{uuu}, \Delta_{vvv}$ at the junction point (u_k, v_l) by Eq.(15), for all $k = 4 \ldots n, l = 4 \ldots m$
(b) Calculate L_{kl} at the junction point (u_k, v_l) by Eq.(16), for all $k = 4 \ldots n, l = 4 \ldots m$
(c) Obtain the global measure G^0 by Eq.(17).
(4) Set: End condition = false
while (End condition = false) *do*
(a) Choose the maximum local fairness measure L_{kl}.
(b) Apply local fairing step at the junction point (u_k, v_l). (At the end of this step, one has C^3 continuity at this junction point).
(c) Calculate new global measure G^j by Eq.(17).
End condition: Global measure:

$$\left|G^{j+1} - G^j\right| < \varepsilon \tag{23}$$

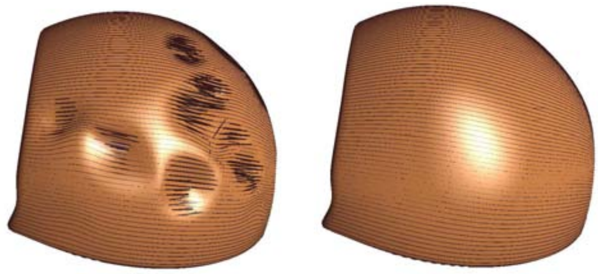
where ε is a very small number defined by the user.
(5) Generate faired B-Spline image surface, isophote, control positions and motion trajectories.



Unfaired Faired

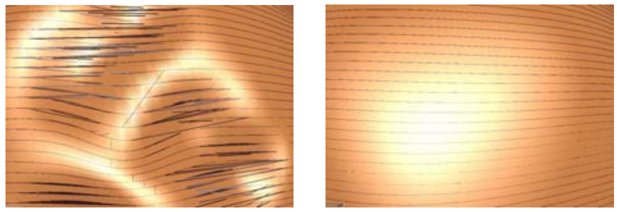
Figure 1. 15x15 Control Net Projected in E^3 for Two-Parameter Bicubic B-Spline Spherical Motion

structure obtained by projecting Quaternion based control structure in E^3, while Figure 2 shows the Quaternion Surface with the isophote before and after tuning. After fine tuning, both the control net and the Quaternion surface appear regular (the wiggles and the dimples in the surface disappear). The isophote lines are continuous indicating satisfactory smoothness(Fig. 3). Since the isophote lines are a function of the direction of light impacting



Unfaired Faired

Figure 2. Quaternion Surface with Isophote for Two-Parameter Bicubic B-Spline Spherical Motion



Unfaired Faired

Figure 3. Isophote details on the Quaternion Surface for Two-Parameter Bicubic B-Spline Spherical Motion

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 and Fig. 5 are the control positions and the trajectory of "teapots", respectively under a uniform bicubic B-Spline Spherical Motion, both before and after fine tuning. As is clear, the control positions after fairing are regularly distributed and the trajectory is better organized. Another indicator of better continuity in Fig. 5 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 the Eq.(19)) 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 $((n-1)*(m-1))$. In this example, we have chosen uniform parameterization for simplicity and faster execution but it is possible to choose centripetal, chord-length, or other parameterizations that map the range behavior more appropriately to the domain (see Farin [1]). The execution time of the algorithm using uniform parameterization for this example was 5 s and it

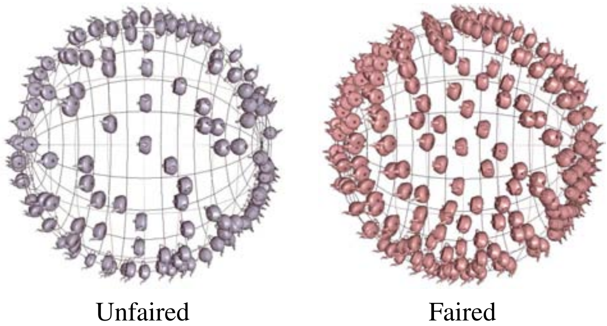


Figure 4. 15x15 Control Positions for Two-Parameter Bicubic B-Spline Spherical Motion

took 1930 iterations for the global fairness measure to reduce from $G_{initial} = 7.84$ to $G_{fair} = 0.002$ on a 2.6 GHz Pentium 4 system with 1 GB 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 tolerance level.

6 FINE TUNING OF TWO-PARAMETER RATIONAL B-SPLINE SPATIAL MOTIONS

A bicubic rational B-Spline image surface is given by Eq. (22). In the spatial case, the dual parts of the quaternions are not zero. The dual parts, $\mathbf{X}^0, \mathbf{d}^0_{i,j}$, involve the translation of the motions.

In the previous section, we discussed the fine tuning algorithm for two parameter B-Spline spatial and spherical motions. The only difference from the spherical case is that for spatial motion fairing, we must consider the real and dual parts of the quaternions simultaneously.

Now, we illustrate the implementation of spatial fine tuning algorithm by a few examples. Since spatial motion involves both the real part and the dual part as given by Eq. (11) and Eq. (12), fine tuning step requires that both the component be fine tuned. Fine tuning of the real part is already illustrated in Sec. (5) via some examples and we use the same value of real part in the succeeding examples. In fine tuning the dual part, there are two choices – one is to fine tune the dual part image surface and the other is to fair the translation part directly, which means fairing the translation image surface. The algorithm is applied identically in both the cases and both yield a fairer, though not identical surface. Here, we only illustrate fine tuning the dual part. Figure (6) shows quaternion surface and isophote for the dual part only. Even though unfaired surface, which is a bicubic B-spline formulation is equipped with C^2 continuity, it is locally irregular. During fine tuning process, the algorithm iteratively removes

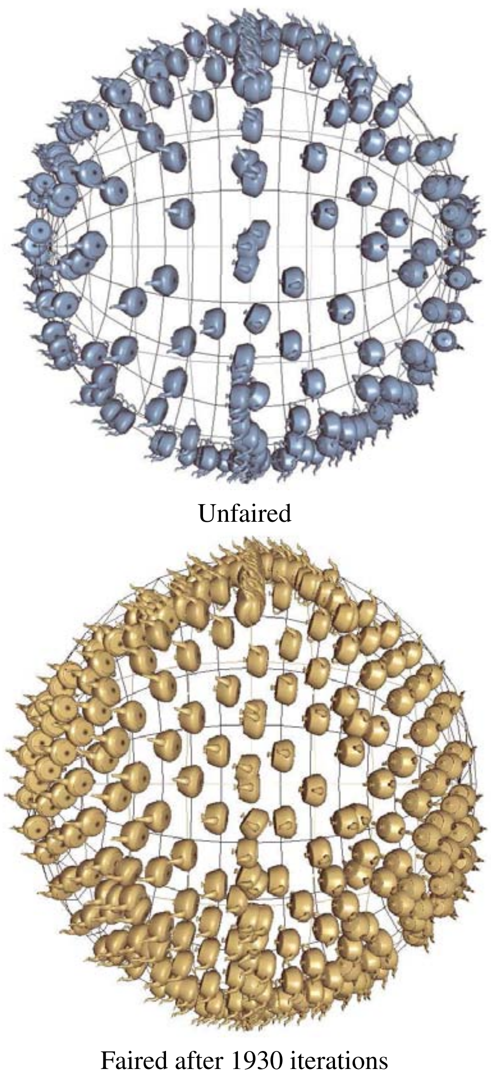


Figure 5. Trajectory under Two-Parameter Bicubic Rational B-Spline Spherical Motion

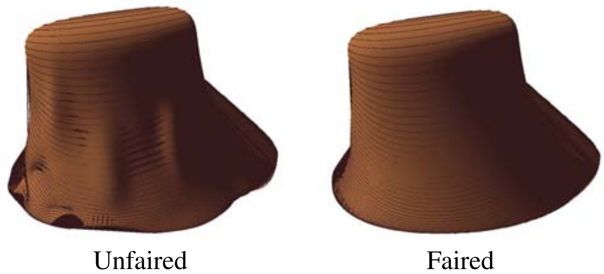


Figure 6. Quaternion Surface and Isophote for Dual Part under Two-Parameter Bicubic Rational B-Spline Spatial Motion

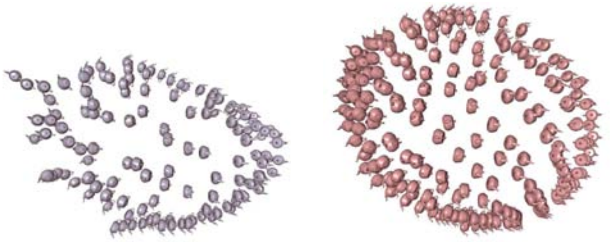


Figure 7. 15x15 Control Positions for Two-Parameter Bicubic B-Spline Spatial Motion

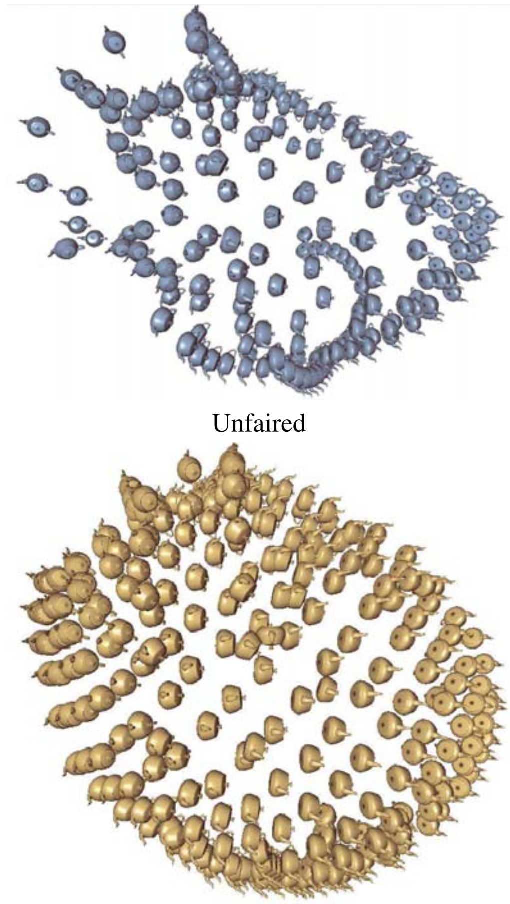


Figure 8. Trajectory under Two-Parameter Bicubic Rational B-Spline Spatial Motion

such imperfections.

Figure 7 and Fig. 8 show the control positions and the trajectory 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 random even though it is a C^2 continuous motion. After fairing, the order in the motion and the control position improves. On the same hardware system, the run time of the algorithm was 8 s (3052 iterations) and it reduced the global fairness measure from $G_{initial} = 10.195$ to $G_{fair} = 0.003$.

7 CONCLUSION

A method is presented for automatic fairing of two-parameter rational B-Spline motion by extending the method for surface fairing 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 5-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 DMI-9800690 to SUNY Stony Brook.

REFERENCES

[1] Farin, G., 1996, *Curves and Surfaces for Computer-Aided Geometric Design: A Practical Guide*, 4th edition, Academic Press

[2] Farin, G., Reinm, G., Sapidis, N., Worsey, A.J., 1987, "Fairing cubic B-spline curves," Computer Aided Geometric Design, Vol. 4, pp. 91-103

[3] Sapidis, N., Farin, G., 1990, "Automatic fairing algorithm for B-spline curves," Computer Aided Design, Vol. 23, pp. 121-129

[4] Kjellander, J.A., 1983, "Smoothing of Bicubic Parametric Splines," Computer Aided Design, Vol. 15, pp. 175-179

[5] Kane, T.R., Levinson, D.A., and Likins, P.W., 1983, *Spacecraft Dynamics*, McGraw-Hill, New York

[6] Fillmore, J.P., 1984, "A Note on Rotation Matrices," IEEE Computer Graphics & Application, Vol. 4, No. 2, pp. 30-33

[7] Roschel, O., 1998, "Rational motion Design - A survey," Computer-Aided Design, Vol. 30, No. 3, pp. 169-178

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

[9] Shoemake, K., 1985, "Animating Rotation with Quaternion Curves," *Proceedings of the 12th annual conference*

on Computer graphics and interactive techniques, ACM Press, pp. 245-254

[10] Nielson, G. M. and Heiland, R.W., 1992, "Animated Rotations using Quaternion and Splines on a 4D Sphere," Programming Comput. Software, Vol. 18, No. pp. 145-154

[11] Nielson, G. M., 2004, "Nu-Quaternion Splines for the Smooth Interpolation of Orientations," IEEE Transactions on Visualization and Computer Graphics, Vol. 10, No. 2, pp. 224-229

[12] Kim, M. S. and Nam, K. W., 1995, "Interpolating Solid Orientations with Circular Blending Quaternion Curves," Computer-Aided Design, Vol. 27, No. 5, pp. 385-398

[13] Kim, Myoung-Jun , Kim, Myung-Soo and Shin, Sung Yong, 1995, "A C^2 Continuous B-spline Quaternion Curve Interpolating a Given Sequence of Solid Orientations," *Proceedings of the Computer Animation*, pp. 72

[14] McCarthy, J.M., 1990, *Introduction to Theoretical Kinematics*, MIT

[15] Ge, Q. J. and Ravani, B., 1994, "Geometric Construction of Bezier Motions," Journal of Mechanical Design, Vol. 116, No. 3, pp. 749-755

[16] Study, E., 1903, *Die Geometrie der Dynamen*, Verlag Teubner

[17] Juttler, B., 1994, "Visualization of Moving-Objects Using Dual Quaternion Curves," Computers & Graphics, Vol. 18, No. 3, pp. 315-326

[18] Juttler, B. and Wagner, M. G., 1996, "Computer-Aided Design with Spatial Rational B-Spline Motions," Journal of Mechanical Design, Vol. 118, No. 2, pp. 193-201

[19] Wagner, M.G., 1994, "A Geometric Approach to Motion Design," Ph.D. dissertation, Technische Universitt Wien

[20] Wagner, M. G., 1995, "Planar Rational B-Spline Motions," Computer-Aided Design, Vol. 27, No. 2, pp. 129-137

[21] Ge, Q.J. and Ravani, B., 1993, "Computational Geometry and Motion Approximation," *Computational Kinematics*, J. Angeles, eds. et al., KAP

[22] Ge, Q. J. and Ravani, B., 1994, "Computer-Aided Geometric Design of Motion Interpolants," Journal of Mechanical Design, Vol. 116, No. 3, pp. 756-762

[23] Ravani, B. and Roth, B., 1984, "Mappings of Spatial Kinematics," Journal of Mechanisms Transmissions and Automation in Design-Transactions of the ASME, Vol. 106, No. 3, pp. 341-347

[24] Purwar, A. and Ge, Q. J., 2005, "On the Effects of Dual Weights in Computer Aided Design of Rational Motions," Journal of Mechanical Design, In Press

[25] Srinivasan, L., and Ge, Q. J., 1998, "Fine tuning of rational B-spline motions," ASME Journal of Mechanical Design, Vol. 120, No. 1, pp. 46-51

[26] Fang, Y. C., Hsieh, C. C., Kim, M. J., Chang, J. J. and Woo, T. C., 1998, "Real Time Motion Fairing with Unit Quaternions," Computer-Aided Design, Vol. 30, No. 3, pp. 191-198

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

[28] 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," *Proc. Symp. Virtual Reality in Manufacturing Research and Education*

[29] Lee, J. and Shin, S., 2002, "General Construction of Time-Domain Filters for Orientation Data," IEEE Transactions on Visualization and Computer Graphics, Vol. 8, No. 2, pp. 119-128

[30] Hsieh, C. C., 2002, "Motion Smoothing Using Wavelets," Journal of Intelligent & Robotic Systems, Vol. 35, No. 2, pp. 157-169

[31] Hsieh, C. C., 2001, "B-Spline Wavelet-based Motion Smoothing," Computers & Industrial Engineering, Vol. 41, No. 1, pp. 59-76

[32] Hsieh, C. C. and Chang, T. Y., 2003, "Motion Fairing using Genetic Algorithms," Computer-Aided Design, Vol. 35, No. 8, pp. 739-749

[33] Angels, J., 1997, *Fundamentals of Robotic Mechanical Systems, Theory, Methods, and Algorithms*, Springer-Verlag, New York

[34] Gupta, K.C., 1997, *Mechanics and Control of Robots*, Springer-Verlag, New York

[35] Ge, Q. J. and Sirchia, M., 1999, "Computer Aided Ggeometric Design of Two-Parameter Freeform Motions," Journal of Mechanical Design, Vol. 121, No. 4, pp. 502-506

[36] Ge, Q. J., 1996, "Kinematics-driven geometric modeling: A framework for simultaneous tool-path generation and sculptured surface design", *Proc. 1996 IEEE Robotics and Automation Conference*, Vol. 2, pp 1819-1824, Minneapolis, MN, April 1996.

[37] Zhang, W., Zhang, Y. F. and Ge, Q. J., 2004, "Five-axis Tool Path Generation for Sculptured Surface Machining using Rational Bëzier Motions", Computer-Aided Design and Applications, ISSN 1686-4360, Vol. 1, No. 2, pp. 251-260.

[38] Kjellander, J.A., 1983, "Smoothing of Bicubic Parametric Surfaces," Computer Aided Design, Vol. 15, pp. 289-293

[39] Nowacki H., Reese, D., 1983, "Design and fairing ship surfaces," In *Surfaces in CAGD*, Barnhill R., Bohm, W.(eds.), pp. 121-134, Amsterdam: North-Holland

[40] Lott, N.J. and Pullin, D.I., 1988, "Method for Fairing B-Spline Surfaces," Computer-Aided Design, Vol. 20, pp. 597-604

[41] Hahmann, S. 1998, "Shape improvement of surfaces," *Computing* [*Suppl*], Vol. 13, pp. 135-152

[42] Welch, W. and Witkin, A., 1992, "Variational Surface Modeling," In *SIGGRAPH '92 Proceedings*, Vol. 26(2), pp. 157-166

[43] Wesselink W.and Veltcamp, R.C., 1995, "Interactive Design of Contrained Variational Curves," Computer-Aided

Geometric Design, Vol. 12, No. 5, pp. 533-546
[44] Greiner, G. and Seidel, H.-P., 1997, "Automatic Modeling of Smooth Spline Surfaces," In *Proc. WSCG '97*, N. Magnenat-Thalmann and V. Skala, editors, pp. 665-675
[45] Poeschl, T., 1984, "Detecting surface irregularities using isophotes," Computer Aided Geometric Design, Vol. 1, pp. 163-168