Surface design based on direct curvature editing

Yuki Kineri\textsuperscript{a}, Shuhei Endo\textsuperscript{b}, Takashi Maekawa\textsuperscript{b,}\textsuperscript{*}

\textsuperscript{a} Ricoh Co., Ltd, Japan
\textsuperscript{b} Department of Mechanical Engineering, Yokohama National University, Japan

HIGHLIGHTS

- Surface design based on direct curvature editing is introduced.
- A point-based curvature control is extended to a curve-based control.
- A log-aesthetic curve is embedded into existing design surfaces.

ARTICLE INFO

Article history:
Received 9 September 2013
Accepted 3 May 2014

Keywords:
Curvature control
B-spline curve and surfaces
Log-aesthetic curves
Cross curvature

ABSTRACT

This paper presents a novel method for modifying the shapes of existing uniform bi-cubic B-spline surfaces by interactively editing the curvatures along isoparametric curves. The method allows us to edit the curvatures of the two intersecting isoparametric curves at each knot with specified positions, unit tangents, and unit normals. The user adjusts the radii of circles, representing the radii of curvature in the \( u \) and \( v \) isoparametric directions directly via a GUI without having to work with control points and knots. Such shape specifications are converted into iterative repositionings of the control points on the basis of geometrical rules. Using these point-based curvature-editing techniques, we successfully embedded log-aesthetic curves into existing surfaces along their isoparametric curves. Moreover, we were able to distribute the cross curvature with log-aesthetic variation along the isoparametric curves. We applied our technique to the design of automobile hood surfaces to demonstrate the effectiveness of our algorithms.

1. Introduction

In the field of structural mechanics, it is well known that a higher stress concentration occurs in regions with a smaller radius of curvature [1]. The presence of such regions causes an object to experience a considerable increase in maximum stress. Therefore, engineers must design the geometry to minimize stress concentration by avoiding high curvatures. Curvature control also plays important roles in fluid dynamics. The surface curvature distribution of airfoils and blades near the leading edge is essential for optimum aerodynamic, thermoeconomic, and overall performance of turbomachinery-based power plants [2].

In aesthetic surface design, a new design can be accelerated by the reuse of existing designs if the B-spline forms are available from previous designs, in whole or part [3]. The designer can interactively make local modifications to the existing B-spline surfaces defined by knots and control points. However, modifying the surfaces to a desired shape via direct manipulation of knots and control points is difficult. Accordingly, researchers have developed tools that allow the user to change shapes of surfaces in an intuitive way and convert them into modifications in control points locations and knot vectors [4]. Séquin [5] envisioned a CAD system, in which a designer specifies boundary conditions and constraints for a car hood surface panel, and then picks a suitable cost functional, from which the system generates a desired surface via optimization.

In this paper, we introduce a novel method for modifying the shape of existing uniform bi-cubic B-spline surfaces by interactively editing the curvatures or fitting to prescribed curvature values along isoparametric curves. The \textit{point-based} curvature control method allows us to edit the curvatures of two isoparametric curves at each knot with specified positions, unit tangents, and unit normals. Users adjust the radii of circles representing the radii of curvature of the \( u \) and \( v \) isoparametric curves via a graphical user interface (GUI), without having to work with control points and knots. Such shape specifications are converted into iterative repositionings of the control points on the basis of geometrical rules. We further extend this point-based curvature control technique to \textit{curve-based} curvature control, where we successfully embedded log-aesthetic curves [6,7] into existing surfaces.

\textsuperscript{*} This paper has been recommended for acceptance by Gershon Elber.

\textsuperscript{*} Corresponding author. Tel.: +81 45 339 3930.
E-mail address: maekawa@ynu.ac.jp (T. Maekawa).

http://dx.doi.org/10.1016/j.cad.2014.05.001
0010-4485/© 2014 Elsevier Ltd. All rights reserved.
The contributions of this paper can be summarized as follows:

- We introduce a novel method for editing curvatures of two isoparametric curves at the knots of uniform bi-cubic B-spline surfaces;
- We prove that changing the curvature of one isoparametric curve at a knot does not affect the curvature of the other intersecting isoparametric curve at the same knot as long as the position at the knot does not change;
- We extend point-based curvature control to curve-based control, which allows us to embed a log-aesthetic curve into the surface together with cross curvature (see Section 4.2 for definition) control.

The rest of the paper is organized as follows. In Section 2, we review the literature on direct surface curvature editing, log-aesthetic curves, and iterative geometric interpolation algorithms. In Sections 3 and 4, we examine point-based and curve-based curvature control, respectively. In Section 5, we apply our technique to the design of automobile hood surfaces to demonstrate the effectiveness of our algorithms. In Section 6, we conclude the paper.

2. Related work

In this section we review the literature on direct surface curvature editing, log-aesthetic curves, and iterative geometric interpolation algorithms.

2.1. Direct surface curvature editing

Andersson [8] developed a tool for directly modifying the curvatures of surfaces by solving non-linear partial differential equations. However, his discussions were confined to surfaces expressed as graphs of real-valued functions over the plane. Furthermore, no practical examples are given in his paper, except for a sketch of procedures.

Du and Qin [9] introduced an integrated approach that combines the partial differential equation (PDE) surfaces, and physics-based modeling techniques to allow users to interactively modify a point, normal, and curvature. The formulation results in non-linear equations, so they did not solve them precisely, but instead approximated the curvature by adjusting the distances between two neighboring points.

Eigensatz and Pauly [10] formulated an optimization framework that allows the user to directly manipulate or preserve positional, metric, and curvature constraints anywhere on the surface of a triangular mesh model.

Nasri et al. [11] presented an algorithm for interpolating curves with prescribed cross curvature on Catmull–Clark subdivision surfaces using polygonal complexes. However, curvature control of the feature curve was not conducted.

2.2. Log-aesthetic curves

In his review paper of CAD tools for aesthetic engineering, Séquin [5] stated that one of the key CAD problems is the embedding of beautiful or fair curves on an optimized surface and described that the most direct connection for drawing a fair curve between two points on a smooth surface is a geodesic curve [12,13]. However, sometimes the geodesic curve may be too restrictive for design purposes, as it is necessary to solve a two-point boundary value problem to connect two points. Accordingly, a designer cannot control the curve, since it is an intrinsic curve on the freeform surface. As an alternative, he suggested a curve on a surface, whose geodesic curvature is either constant or linearly varying as a function of arc-length.

Harada et al. [14] showed that many aesthetic curves in nature and in artificial objects exhibit monotonically varying curvature, and hence their logarithmic curvature histograms (LCH) can be approximated by straight lines. The shapes of these curves depend on the slope of LCH, so they are called log-aesthetic curves [6]. We examined the shape of a butterfly wing, and a Japanese sword blade. Fig. 1(a) depicts the common bluebottle swallowtail or blue triangle butterfly whose scientific name is Graphium sarpedon, while Fig. 1(c) shows its logarithmic curvature graph, which is defined in (32) corresponding to the red curve indicated in (b). The straight line in (c) is a least squares fit having a slope of $\alpha = -0.44$. Fig. 2(a)–(c) illustrates the Japanese sword blade, its boundary curves extracted by the computer vision techniques, and the logarithmic curvature graph corresponding to the red portion in (b), respectively. It is clear from Fig. 2(c) that the logarithmic curvature graph is almost a straight line having a slope of $\alpha = -0.99$. Mathematical details of the logarithmic curvature graph are discussed in Section 4.3.

Furthermore, LCH of a clothoid curve (also called an Euler spiral or the spiral of Cornu) and the logarithmic spiral are straight lines with slopes of $1$ and $-1$, respectively. The general formula for a log-aesthetic curve as a function of arc-length was derived by Miura [16]. Recently, Ziatdinov et al. [17] introduced analytic parametric equations for log-aesthetic curves consisting of trigonometric and incomplete gamma functions. Yoshida et al. [7] extended log-aesthetic planar curves to log-aesthetic space curves.

2.3. Iterative geometric interpolation algorithm

Recently, in contrast to the standard surface fitting methods, iterative geometric fitting methods that do not require the solution of a linear system have received attention [18–23]. These methods employ a surprisingly simple geometric-based algorithm which iteratively updates the control points in a global manner based
on a local point surface distance computation and a repositioning procedure.

Iterative geometric interpolation algorithms were introduced by Yamaguchi [18] for uniform cubic B-spline curves and surfaces; Böhm et al. [19] for triangular splines; Lin et al. [20] for nonuniform cubic B-spline curves and surfaces; Fan et al. [21] for Doo–Sabin subdivision surfaces, and Maekawa et al. [22] for Loop and Catmull–Clark subdivision surfaces. Lin [23] proved a conjecture proposed by Maekawa et al. [22] that repositioning the control points parallel to the position error vectors interpolates the input points. Kineri et al. [24] extended the iterative geometric interpolation algorithm to preserve the reflective symmetry of the model while fitting the surface. This technique is important in fitting industrial products such as airplanes, trains, and automobiles that possess reflective symmetry.

A recent study evolved the iterative geometric interpolation algorithm to interpolate a sequence of data points, unit tangent vectors, and curvature vectors [25–27].

3. Point-based curvature control

This section together with Section 4, constitutes the main body of this paper. In this section, we introduce point-based curvature control, and in Section 4 we study curve-based curvature control. Both curvature control methods extend the technique of interpolating a sequence of data points under tangent vectors and curvature vectors constraints [27] in order to directly edit the curvatures of the two intersecting isoparametric curves at knots of the uniform bi-cubic B-spline surfaces. Furthermore, in Section 3.3 we show that the curvature of the u-direction feature curve is not affected by the modification of the u-direction feature curve and vice versa as long as the position of the point of interest on the surface remains unchanged during the modifications. The input to our algorithm is an unclamped uniform bi-cubic B-spline surface. If the input surface is clamped, we can unclamp it using the unclamping algorithm described in [4]. If the knot spacing is not uniform, we can adjust the knot vectors such that the spacing becomes uniform by knot insertion [4].

3.1. Isoparametric curves as feature curves

Let us consider a uniform bi-cubic B-spline surface \( \mathbf{r}(u, v) \) with control points \( \mathbf{P}_{ij} \) where \( 0 \leq i \leq n \) and \( 0 \leq j \leq m \). There are \( (n-2) \times (m-2) \) patches in this surface. The basis functions of uniform B-splines are based on a knot sequence that has uniform spacing, and hence they are not functions of knots. We describe the \((i, j)\)-th uniform bi-cubic B-spline patch \( \mathbf{r}_{ij}(u, v) \) shown in Fig. 3 as

\[
\mathbf{r}_{ij}(u, v) = \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} [M] [\mathbf{P}]_{ij} [M]^T \begin{bmatrix} 1 \\ u \\ u^2 \\ u^3 \end{bmatrix},
\]

where \( 1 \leq i \leq n-2, 1 \leq j \leq m-2 \).

\[
[M] = \begin{bmatrix} 1 & 1 & 4 & 1 & 0 \\ 0 & -3 & 0 & 3 & 0 \\ 0 & 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 & 0 \end{bmatrix},
\]

and the control points \([\mathbf{P}]_{ij}\) are defined as

\[
[\mathbf{P}]_{ij} = \begin{bmatrix} \mathbf{P}_{i-j-1} & \mathbf{P}_{i-j} & \mathbf{P}_{i-j+1} & \mathbf{P}_{i-j+2} \\ \mathbf{P}_{j-1} & \mathbf{P}_{j} & \mathbf{P}_{j+1} & \mathbf{P}_{j+2} \\ \mathbf{P}_{i+j-1} & \mathbf{P}_{i+j} & \mathbf{P}_{i+j+1} & \mathbf{P}_{i+j+2} \\ \mathbf{P}_{i+j+1} & \mathbf{P}_{i+j+2} & \mathbf{P}_{i+j+3} & \mathbf{P}_{i+j+4} \end{bmatrix}.
\]

The \( v = 0 \) isoparametric curve, which we call the u-direction feature curve, is depicted in Fig. 3(a) and is given by

\[
\mathbf{r}_{ij}(u, 0) = \frac{1}{6} \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} [M] [\mathbf{P}]_{ij} \begin{bmatrix} 1 \\ 4 \\ 1 \\ 0 \end{bmatrix}.
\]

Since the last element of the right-most column vector is zero, the \( 4 \times 4 \) control points reduce to \( 4 \times 4 \) as follows:

\[
\mathbf{r}_{ij}(u, 0) = \frac{1}{6} \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} [M] 
\times \begin{bmatrix} \mathbf{P}_{i-j-1} & \mathbf{P}_{i-j} & \mathbf{P}_{i-j+1} & \mathbf{P}_{i-j+2} \\ \mathbf{P}_{j-1} & \mathbf{P}_{j} & \mathbf{P}_{j+1} & \mathbf{P}_{j+2} \\ \mathbf{P}_{i+j-1} & \mathbf{P}_{i+j} & \mathbf{P}_{i+j+1} & \mathbf{P}_{i+j+2} \\ \mathbf{P}_{i+j+1} & \mathbf{P}_{i+j+2} & \mathbf{P}_{i+j+3} & \mathbf{P}_{i+j+4} \end{bmatrix} \begin{bmatrix} 1 \\ 4 \\ 1 \\ 0 \end{bmatrix}.
\]

\[
= \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} [M] \begin{bmatrix} S_{i-j-1} \\ S_{i-j} \\ S_{i+j-1} \\ S_{i+j} \end{bmatrix},
\]

where \( S_{ij} \) are the control points of the u-direction feature curve given by

\[
S_{ij} = \frac{1}{6} (\mathbf{P}_{i-j-1} + 4\mathbf{P}_{i-j} + \mathbf{P}_{i-j+1}).
\]
Nasri [28] calls the $4 \times 3$ matrix of (5) the polygonal complex to indicate simply a polygonal mesh whose limit of binary subdivision is a curve rather than a surface. Thick lines in Fig. 3(a) and (b) represent the polygonal complex.

Similarly, the control points of the $v$-direction feature curve $T_{ij}$, which is depicted in Fig. 3(b), can be obtained as follows:

$$T_{ij} = \frac{1}{6} \left( P_{i-1,j} + 4P_{ij} + P_{i+1,j} \right).$$  \hspace{1cm} (8)

### 3.2 Direct modification of curvatures at knots

Suppose that we are given an existing B-spline surface, and want to change the curvature of the isoparametric curves at knots $r_i(0,0)$. In other words, we are given the position vector $r_i(0,0)$, the unit tangent $t_i^u$ or $t_i^v$, and the curvature vector $\kappa_i^u n_i^v$ or $\kappa_i^v n_i^u$ at knots, where superscripts $u$ and $v$ denote the direction of the feature curves, and $\kappa_i^u$, $\kappa_i^v$ and $n_i^u$, $n_i^v$ are the curvatures and unit normal vectors, respectively. We control the curvature values $\kappa_i^u$ or $\kappa_i^v$ of the feature curve. Our user interface uses the CallBack function of AntTweakBar, an intuitive GUI library, to adjust the magnitude of the radius of curvature of the isoparametric curve as shown in Fig. 4. The numeric value of the curvature shown in the dialog box is increased or decreased by turning the circle of the RotoSlider of the AntTweakBar library. Let us consider the control of $\kappa_i^u$, the control of $\kappa_i^v$ can be performed similarly. First, we project the three consecutive control points of the $u$-direction feature curve $S_{i-1,j}, S_i,j, S_{i+1,j}$ onto the osculating plane $A$ spanned by $t_i^u$ and $n_i^u$ at $r_i(0,0)$, and denote them as $\hat{S}_{i-1,j}, \hat{S}_i,j, \hat{S}_{i+1,j}$. Then, we have

$$\hat{S}_{i-1,j} = S_{i-1,j} - (\langle S_{i-1,j} - r_i(0,0) \rangle \cdot b_i^u) b_i^u,$$

$$\hat{S}_i,j = S_i,j - (\langle S_i,j - r_i(0,0) \rangle \cdot b_i^v) b_i^v,$$

$$\hat{S}_{i+1,j} = S_{i+1,j} - (\langle S_{i+1,j} - r_i(0,0) \rangle \cdot b_i^v) b_i^v,$$

where $b_i^u = t_i^u \times n_i^u$. We interactively adjust the curvature of the isoparametric curve by changing the radius of the circle (see Fig. 4), which represents the radius of curvature $1/\kappa_i^u$, via the following equations:

$$\hat{S}_{i-1,j} = \hat{S}_{i-1,j} + \frac{1}{2} t_i^u \left( \| \hat{S}_{i-1,j} - \hat{S}_{i-1,j} \|^2 - \frac{12}{\kappa_i^u} (\langle r_i(0,0) - \hat{S}_{i-1,j} \rangle \times t_i^u) \cdot b_i^v \right).$$

$$\hat{S}_{i+1,j} = \hat{S}_{i+1,j} + \frac{1}{2} t_i^v \left( \| \hat{S}_{i+1,j} - \hat{S}_{i+1,j} \|^2 - \frac{12}{\kappa_i^v} (\langle r_i(0,0) - \hat{S}_{i+1,j} \rangle \times t_i^v) \cdot b_i^u \right).$$

Fig. 3. (i, j)-th uniform bi-cubic B-spline surface $r_i(u, v)$ and its corresponding control net (green): (a) The red curve and the red balls represent $u$-direction feature curve ($v = 0$), and its control points $S_i,j$, respectively. (b) The blue curve and the blue balls represent $v$-direction feature curve ($u = 0$), and its control points $T_{ij}$, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Fig. 4. Our user interface for point-based curvature control uses the CallBack function of AntTweakBar to adjust the radii of red and blue circles representing the radii of curvature of $u$ and $v$ isoparametric curves, respectively. (a) Before modification. (b) After the modification of curvature in the $v$-direction from $-1.63$ to $-4.22$. Note that, the curvature in the $u$-direction $-6.67$ remains the same. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$\tilde{S}_{i,j} = \frac{3}{2} r_i(0,0) - \frac{1}{4} \hat{S}_{i-1,j} - \frac{1}{4} \hat{S}_{i+1,j} \| \hat{S}_{i-1,j} - \hat{S}_{i+1,j} \| t_i^u,$$

$$\tilde{S}_{i+1,j} = \hat{S}_{i+1,j} + \frac{1}{2} t_i^v \left( \| \hat{S}_{i+1,j} - \hat{S}_{i-1,j} \|^2 + \frac{12}{\kappa_i^v} (\langle r_i(0,0) - \hat{S}_{i+1,j} \rangle \times t_i^v) \cdot b_i^u \right),$$

where $\tilde{S}_{i-1,j}, \tilde{S}_i,j, \tilde{S}_{i+1,j}$ are the control points after editing. The proof of convergence of this algorithm is given in [27]. In practice, it is recommended that $\tilde{S}_{i-1,j}, \tilde{S}_i,j, \tilde{S}_{i+1,j}$ are adjusted iteratively so that $\| \tilde{S}_i,j - \tilde{S}_{i-1,j} \| = \| \tilde{S}_i,j - \tilde{S}_{i+1,j} \|$ to avoid unwanted artifacts [27].

Once the control points of the feature curve are updated, we feedback the changes to the surface control points. If we denote...
the repositioning vectors by \( \delta_{ij} = \tilde{S}_{ij} - S_{ij} \), where \( \lambda = i - 1, i, i + 1, \) and using (7), we have
\[
\tilde{S}_{ij} = S_{ij} + \delta_{ij} \quad \quad \quad (15)
\]
\[
= \frac{1}{6} \left( (P_{ij+1} - 6S_{ij}) + 4(P_{ij} + \delta_{ij}) + (P_{ij+1} + \delta_{ij}) \right). \quad \quad \quad (16)
\]
Accordingly, the new surface control points can be written as
\[
\tilde{P}_{ij+1} = P_{ij+1} + \delta_{ij}, \quad \quad \quad (17)
\]
\[
\tilde{P}_{ij} = P_{ij} + \delta_{ij}, \quad \quad \quad (18)
\]
\[
\tilde{P}_{ij+1} = P_{ij+1} + \delta_{ij}. \quad \quad \quad (19)
\]
In some situations, the user may want to change the positions \( r_i(0, 0) \) at knots along the feature curve to \( V_i \). In those cases we iteratively construct the sequence \( S_{ij}^{(k+1)} \) [18,27]:
\[
S_{ij}^{(k+1)} = \frac{1}{4} (6V_{ij} - S_{ij}^{(k)} - S_{ij}^{(k+1)}). \quad \quad \quad (20)
\]
Farin [29] stated that this method amounts to solving the system by Gauss–Seidel iteration, and that it will always converge.

3.3. Independence of curvature control between two intersecting isoparametric curves

In the following, we show that the curvature of the \( v \)-direction feature curve is not affected by the modification of the \( u \)-direction feature curve and vice versa as long as the position of \( r_k(0, 0) \) remains unchanged during the modifications. Let us denote the control points of the curvature of the \( u \)-direction feature curve after modification by \( S_{ij} \).

Since the position \( r_k(0, 0) \) does not change before and after the curvature editing process, we can write
\[
r_k(0, 0) = \frac{1}{6} (S_{i-1,j} + 4S_{ij} + S_{i+1,j})
\]
\[
= \frac{1}{6} (\tilde{S}_{i-1,j} + 4\tilde{S}_{ij} + \tilde{S}_{i+1,j}). \quad \quad \quad (21)
\]
Using the repositioning vector \( \delta_{ij} = \tilde{S}_{ij} - S_{ij} \) (21) can be rewritten as follows:
\[
\delta_{i-1,j} + 4\delta_{ij} + \delta_{i+1,j} = 0. \quad \quad \quad (22)
\]
Based on (8), the control points of the \( v \)-direction feature curve \( \tilde{T}_{ij} \) can be expressed in terms of the control points of the modified surface \( \tilde{P} \):
\[
\tilde{T}_{ij} = \frac{1}{6} \left( \tilde{P}_{i-1,j} + 4\tilde{P}_{ij} + \tilde{P}_{i+1,j} \right)
\]
\[
= \frac{1}{6} \left( \tilde{P}_{i-1,j} + \delta_{i-1,j} + 4(\tilde{P}_{ij} + \delta_{ij}) + \tilde{P}_{i+1,j} + \delta_{i+1,j} \right)
\]
\[
= T_{ij} + \frac{1}{6} (\delta_{i-1,j} + 4\delta_{ij} + \delta_{i+1,j})
\]
\[
= T_{ij}. \quad \quad \quad (23)
\]
Similarly we can show that \( \tilde{T}_{ij-1} = T_{ij-1} \) and \( \tilde{T}_{ij+1} = T_{ij+1} \). This clearly shows that the process of curvature control of the \( u \)-direction feature curve does not affect the control points of the \( u \)-direction feature curve. In other words, we are able to control the curvatures in both directions independently, which is an important property in surface design. The curvature of \( u \) and \( v \) isoparametric curves shown in Fig. 4(a) are \(-6.67 \) and \(-1.63\), respectively. Fig. 4(b) clearly shows that despite the curvature modification in the \( u \)-direction from \(-1.63 \) to \(-2.22\), the curvature in the \( u \)-direction \(-6.67 \) remains the same after the modification.

4. Curve-based curvature control

The previous section discussed point-based curvature control, but there are many aesthetic design applications that require curvature control along the feature curves. In this section, we introduce a novel interactive method for embedding log-aesthetic curves into the surface as feature curves.

4.1. Formulation

Curvature control of the feature curve requires increasing the curve’s degree of freedom [27]. This can be done by inserting two knots in each knot span of the u-knot vector alone, the v-knot vector alone, or both the u and v knot vectors. For the sake of generality, we study the case of ternary subdivision in both directions [30] in the Appendix. From (A.4) the \( u \)-direction feature curve of the ternary subdivided patch is given by
\[
r_{ij}(u, 0) = \left[ \begin{array}{ccc} 1 & u & u^2 \end{array} \right] \begin{bmatrix} S_{ij-1}^{-1} & S_{ij}^{-1} & S_{ij+1}^{-1} \\ S_{ij-1} & S_{ij} & S_{ij+1} \\ S_{ij-1} & S_{ij} & S_{ij+1} \end{bmatrix}, \quad \quad \quad (24)
\]
where \( k = 3i, l = 3j \), and
\[
S_{ij} = \frac{1}{6} (R_{ij-1} + 4R_{ij} + R_{ij+1}) . \quad \quad \quad (25)
\]
Replacing the subscripts \( i \) and \( j \) by \( k \) and \( l \), respectively, in related equations, and \( P_{ij} \) by \( R_{ij} \), we are able to apply the point-based curvature control (12)–(14) to the red control points grouped by the dashed line as shown in Fig. 5.

4.2. Cross curvature control

Suppose that we have a \( u \)-direction feature curve \( r_k(u, 0) \); there exists a \( u = 0 \) isoparametric curve, which intersects the feature curve at \( r_k(0, 0) \). We call the curvature of the \( u = 0 \) isoparametric curve at \( r_k(0, 0) \) as cross curvature and vice versa. Nasri et al. [11] controlled the curvature of the cross-section of the limit surface of the Catmull–Clark surface by the plane perpendicular to the limit curve at the limit point corresponding to the center control point. This is not the same as the curvature of the other isoparametric curve, which can be either smaller (if it crosses the controlled path at other than a right angle), or larger (if its osculating plane does not contain the surface normal) [11].

We can add further emphasis to the feature curve by imposing cross curvature control along it. Suppose that we have a uniform bi-cubic B-spline surface with \((n + 1) \times (m + 1)\) control points. Replacing \( S \) by \( T \) and the related subscripts of (9) to (19), we are able to obtain the control points of the \( v \)-direction feature curves \( T_{ij}, i = 1, \ldots, n - 1, \lambda = j - 1, j, j + 1, \) which can be used to obtain the boundary control points \( P_{ij-1}, P_{ij+1}, i = 1, \ldots, n - 1, \lambda = j - 1, j, j + 1, \) by solving the following tridiagonal matrix system:
\[
[4 1 0 0 \cdots 0 1 4 1 0 \cdots 0 1 4 1 \cdots 0] \begin{bmatrix} P_{1,1} \\ P_{2,1} \\ \vdots \\ P_{n-1,1} \end{bmatrix} = \begin{bmatrix} T_{1,1} - \frac{P_{0,1}}{6} \\ T_{2,1} \\ \vdots \\ T_{n-1,1} - \frac{P_{n-1,1}}{6} \end{bmatrix}, \quad \quad \quad (26)
\]
where the boundary control points \( P_{0,1} \) and \( P_{n,1} \) are obtained using (9) to (19). It is well known that a tridiagonal matrix system can be obtained in \( O(n) \) operations [31]. Similarly the control points of the \( u \)-direction feature curves \( S_{ij} \) can be used in (26) to obtain the surface control points \( P_{ij-1}, P_{ij}, P_{ij+1} \), where \( \lambda = i - 1, i, i + 1, j = 1, \ldots, m - 1 \).
4.3. Log-aesthetic curves

4.3.1. Log-aesthetic planar curve

A general formula for log-aesthetic curve was defined by Miura [16] as follows:

$$\log \left( \frac{ds}{d\rho} \right) = \alpha \log \rho + c,$$

(27)

where $\rho$ is the radius of curvature of the curve, $\alpha$ and $c$ are the slope and the $y$-intercept, respectively of the straight line in the double logarithmic graph as shown in Fig. 6. It is known that the slopes of a clothoid, a Nielsen’s spiral, a logarithmic spiral, the involute of a circle, and a circle are $-1, 0, 1, 2, \infty$, respectively [17].

It is easy to derive from (27) that

$$\frac{ds}{d\rho} = \rho^{\alpha-1} \Lambda,$$

(28)

where $\Lambda = e^{-c}$.

Let the reference point $P_r$ be any point on the curve except for the point whose radius of curvature is either 0 or $\infty$. The following constraints are placed at the reference point (see Fig. 7) [17]:

- scaling: $\rho = 1$ at $P_r$, which means that $s = 0$ and $\theta = 0$ at the reference point;
- translation: $P_r$ is placed at the origin of the Cartesian coordinate system;
- rotation: the tangent line to the curve at $P_r$ is parallel to the $x$-axis.

Substituting the following relation between curvature $\kappa$ and tangent angle $\theta$

$$\kappa = \frac{1}{\rho} = \frac{d\theta}{ds},$$

(29)

into (28), we obtain

$$\frac{d\theta}{d\rho} = \frac{ds}{\rho d\rho} = \rho^{\alpha-2} \Lambda.$$

(30)

Integrating (30) with respect to $\theta$ yields

$$\rho(\theta) = \begin{cases} e^{\alpha \theta}, & \alpha = 1 \\ \left( (\alpha - 1) \Lambda \theta + 1 \right)^{-\frac{1}{\alpha - 1}}, & \text{otherwise,} \end{cases}$$

(31)

or

$$\kappa(\theta) = \begin{cases} e^{-\alpha \theta}, & \alpha = 1 \\ \left( (\alpha - 1) \Lambda \theta + 1 \right)^{-\frac{1}{\alpha - 1}}, & \text{otherwise.} \end{cases}$$

(32)

From geometry we have

$$dx = \cos \theta ds, \quad dy = \sin \theta ds,$$

(33)

and $ds = \rho d\theta$ from (29), the parametric equation for the log-aesthetic curve is given as follows [17]:

$$x(\psi) = \int_0^\psi \rho(\theta) \cos(\theta)d\theta,$$

(34)

$$y(\psi) = \int_0^\psi \rho(\theta) \sin(\theta)d\theta.$$  

(35)
known as the Frenet–Serret system.

The arclength, the points on the log-aesthetic space can be obtained by solving the system of first order differential equations (36) and (37) are called type 1 curves, while those defined by (36) and (38) are called type 2 curves. The curvature parameter \( \Lambda \) and the torsion \( \nu \) at the reference points are obtained through minimization [6]. In a manner similar to that of the planar curve case, we obtain curvatures and torsions for the log-aesthetic space curves as follows:

\[
\kappa(s) = \begin{cases} e^{-As}, & \alpha = 1 \\ (A\alpha s + 1)^{-\frac{1}{2}}, & \text{otherwise} \end{cases}
\]

\[
\tau(s) = \begin{cases} e^{-(\Omega s + \log \beta)}, & \beta = 0 \\ (\Omega \beta s + 1)^{-\frac{1}{2}}, & \text{otherwise} \end{cases}
\]

\[
\tau(s) = \begin{cases} e^{-(\Omega s + \log \beta)}, & \beta = 0 \\ (-\Omega \beta s + 1)^{-\frac{1}{2}}, & \text{otherwise}. \end{cases}
\]

Log-aesthetic space curves are defined as curves whose functions of the radius of curvature and the radius of torsion are given by (36) and either (37) or (38), respectively. Curves defined by the intrinsic equations (36) and (37) are called type 1 curves, while those defined by (36) and (38) are called type 2 curves. The radius of curvature and the radius of torsion of type 1 curves increase monotonically with respect to arc length. In contrast, the radius of curvature of type 2 curves increases monotonically whereas their radius of torsion decreases monotonically.

If we denote the prime ‘ \( \prime \) by the differentiation with respect to the arc length, the points on the log-aesthetic space can be obtained by solving the system of first order differential equations known as the Frenet–Serret system

\[
t'(s) = \kappa(s)n, \quad n' = -\kappa(s)t + \tau(s)b, \quad b' = -\tau(s)n, \quad (39)
\]

together with \( \mathbf{r}' = \mathbf{r}(s) \) as an initial value problem, where initial values at \( s = 0 \) are given by

\[
\mathbf{r}(0) = [0, 0, 0]^T, \quad \mathbf{t}(0) = [1, 0, 0]^T, \quad \mathbf{n}(0) = [0, 1, 0]^T, \quad \mathbf{b}(0) = \mathbf{t}(0) \times \mathbf{n}(0).
\]

Log-aesthetic space curves are defined as curves whose functions of the radius of curvature and the radius of torsion are given by (36) and either (37) or (38), respectively. Curves defined by the intrinsic equations (36) and (37) are called type 1 curves, while those defined by (36) and (38) are called type 2 curves. The curvature parameter \( \Lambda \) and the torsion \( \nu \) at the reference points are obtained through minimization [6]. In a manner similar to that of the planar curve case, we obtain curvatures and torsions for the log-aesthetic space curves as follows:

\[
\kappa(s) = \begin{cases} e^{-As}, & \alpha = 1 \\ (A\alpha s + 1)^{-\frac{1}{2}}, & \text{otherwise} \end{cases}
\]

\[
\tau(s) = \begin{cases} e^{-(\Omega s + \log \beta)}, & \beta = 0 \\ (\Omega \beta s + 1)^{-\frac{1}{2}}, & \text{otherwise} \end{cases}
\]

\[
\tau(s) = \begin{cases} e^{-(\Omega s + \log \beta)}, & \beta = 0 \\ (-\Omega \beta s + 1)^{-\frac{1}{2}}, & \text{otherwise}. \end{cases}
\]

Log-aesthetic space curves are defined as curves whose functions of the radius of curvature and the radius of torsion are given by (36) and either (37) or (38), respectively. Curves defined by the intrinsic equations (36) and (37) are called type 1 curves, while those defined by (36) and (38) are called type 2 curves. The radius of curvature and the radius of torsion of type 1 curves increase monotonically with respect to arc length. In contrast, the radius of curvature of type 2 curves increases monotonically whereas their radius of torsion decreases monotonically.

If we denote the prime ‘ \( \prime \) by the differentiation with respect to the arc length, the points on the log-aesthetic space can be obtained by solving the system of first order differential equations known as the Frenet–Serret system

\[
t'(s) = \kappa(s)n, \quad n' = -\kappa(s)t + \tau(s)b, \quad b' = -\tau(s)n, \quad (39)
\]

together with \( \mathbf{r}' = \mathbf{r}(s) \) as an initial value problem, where initial values at \( s = 0 \) are given by

\[
\mathbf{r}(0) = [0, 0, 0]^T, \quad \mathbf{t}(0) = [1, 0, 0]^T, \quad \mathbf{n}(0) = [0, 1, 0]^T, \quad \mathbf{b}(0) = \mathbf{t}(0) \times \mathbf{n}(0).
\]

Fig. 8. Interactive embedding of log-aesthetic curves which are depicted in red. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

4.3.2. Log-aesthetic space curve

Yoshida et al. [7] presented an interactive control method for log-aesthetic space curves whose logarithmic curvature and torsion graphs are both linear. The linearity of these graphs imposes the constraint that the curvature and torsion be monotonically varying. They determine the curve segment of the log-aesthetic curve by specifying the positions of the two endpoints, their tangents, the slopes, \( \alpha \) and \( \beta \) of the straight lines of the logarithmic curvature and torsion graphs. The curvature parameter \( \Lambda \) and the torsion \( \nu \) at the reference points are obtained through minimization [6]. In a manner similar to that of the planar curve case, we obtain curvatures and torsions for the log-aesthetic space curves as follows:

\[
\kappa(s) = \begin{cases} e^{-As}, & \alpha = 1 \\ (A\alpha s + 1)^{-\frac{1}{2}}, & \text{otherwise} \end{cases}
\]

\[
\tau(s) = \begin{cases} e^{-(\Omega s + \log \beta)}, & \beta = 0 \\ (\Omega \beta s + 1)^{-\frac{1}{2}}, & \text{otherwise} \end{cases}
\]

\[
\tau(s) = \begin{cases} e^{-(\Omega s + \log \beta)}, & \beta = 0 \\ (-\Omega \beta s + 1)^{-\frac{1}{2}}, & \text{otherwise}. \end{cases}
\]

Log-aesthetic space curves are defined as curves whose functions of the radius of curvature and the radius of torsion are given by (36) and either (37) or (38), respectively. Curves defined by the intrinsic equations (36) and (37) are called type 1 curves, while those defined by (36) and (38) are called type 2 curves. The radius of curvature and the radius of torsion of type 1 curves increase monotonically with respect to arc length. In contrast, the radius of curvature of type 2 curves increases monotonically whereas their radius of torsion decreases monotonically.

If we denote the prime ‘ \( \prime \) by the differentiation with respect to the arc length, the points on the log-aesthetic space can be obtained by solving the system of first order differential equations known as the Frenet–Serret system

\[
t'(s) = \kappa(s)n, \quad n' = -\kappa(s)t + \tau(s)b, \quad b' = -\tau(s)n, \quad (39)
\]

together with \( \mathbf{r}' = \mathbf{r}(s) \) as an initial value problem, where initial values at \( s = 0 \) are given by

\[
\mathbf{r}(0) = [0, 0, 0]^T, \quad \mathbf{t}(0) = [1, 0, 0]^T, \quad \mathbf{n}(0) = [0, 1, 0]^T, \quad \mathbf{b}(0) = \mathbf{t}(0) \times \mathbf{n}(0).
\]

4.4. Interactive embedding of log-aesthetic curves

The designer first determines the shape of the feature curve through (29) and (35) by selecting the slope \( \alpha, \gamma \)-intercept \( c \) of the straight line in the double logarithmic graph, and the tangent angles at the starting point and the ending point of the feature curve, i.e. \( \theta_1 \) and \( \theta_2 \), adjusting the RotoSlider of the AntTweakBar library interactively as shown in Fig. 8.

The ratio of the length between the starting point and the ending point of the feature curve to that of the normalized log-aesthetic curve will be the scale \( S_c \). The positions of the feature curve can be computed from (34) and (35) using some numerical integration scheme, e.g. Gaussian quadrature, followed by the multiplication of the scale factor \( S_c \).

5. Examples

In this section, we give an example of point-based curvature control and two examples of embedding log-aesthetic curves into an automobile hood design.

5.1. Point-based curvature control

Fig. 9(a) shows an automobile wheel well, represented by a uniform bi-cubic B-spline surface of \( 5 \times 8 \) control points, consisting of ten patches. We apply point-based curvature control at two locations of the surface where the magnitude of Gaussian curvature values are large, namely 0.05 and -0.03 as shown in Fig. 10(b). Fig. 9(b) shows the two osculating circles adjusting the radii of curvatures for an elliptic point with Gaussian curvature value of 0.05. We modify the curvatures of the two intersecting isoparametric curves at the knot from -0.20 to -0.18 in the \( u \)-direction, and from -0.32 to -0.30 in the \( v \)-direction in order to reduce the high curvature value yielding Gaussian curvature value of 0.04 to avoid stress concentration. Similarly we reduce the curvature value of the hyperbolic point from -0.08 to -0.06 in the \( u \)-direction, and from 0.41 to 0.30 in the \( v \)-direction yielding Gaussian curvature value of -0.02. Fig. 10(a) and (b) shows the wheel well and its Gaussian curvature distribution before the curvature modifications, and (c) and (d) after the curvature modifications. Here we note that the curvatures of the two isoparametric curves passing through the point are generally not equal to the principal curvatures of the surface at the point. This is the reason why the product of the two curvatures of the isoparametric lines does not agree with the Gaussian.
curvature. However, if the magnitude of the curvatures of the two isoparametric curves decreases, it is obvious that the magnitude of the Gaussian curvature also decreases. This is illustrated by the changes in the blue and red colors turning from the dark color (see Fig. 10(b)) to the light color (see Fig. 10(d)), which implies that the point-based curvature control is successful.

5.2. Curve-based curvature control

5.2.1. Planar feature curve

The automobile hood B-spline surface consisting of $11 \times 11$ control points shown in Fig. 11(a) has a feature curve at its center, so the feature curve is a planar curve. The curvature plot as well as the osculating circles of the cross curvature along the feature curve are depicted in Fig. 11(b) and (c), respectively. The designer first determines the shape of the feature curve through (29) and (35) by selecting the slope $\alpha$ and y-intercept $c$ of the straight line in the double logarithmic graph, and the tangent angles at the starting point and the ending point of the feature curve, i.e. $\theta_s$ and $\theta_e$. The ratio of the length between the starting point and the ending point of the feature curve to that of the normalized log-aesthetic curve will be the scale $S_c$. The positions of the feature curve can be computed from (34) and (35) using some numerical integration scheme, e.g. Gaussian quadrature, followed by the multiplication of the scale factor $S_c$. We divide the feature curve equally by the number of spans, and evaluate the positions, unit tangents, and curvature vectors at the division points. We first interpolate the positions using (20). Since all the calculations are performed in standard form, coordinate transformation must be performed such that the position and orientation of the aesthetic curve match at the starting point of the isoparametric feature curve.

Then, we insert two knots in each span along the feature curve direction, described in the Appendix, to increase the degree of freedom in order to apply the curve-based curvature control technique. Fig. 12(a) illustrates the log-aesthetic curve with $\alpha = 0.5$, $c = 0.5$, $\theta_s = 0.0$ [rad] and $\theta_e = 1.2$ [rad], which is embedded into the original surface shown in Fig. 11. Fig. 12(b) is the modified surface. The curvature plot along the feature curve shown in Fig. 12(c) clearly shows that the feature curve is a monotonically varying log-aesthetic curve, whereas that of the original surface shown in Fig. 11(b) varies without any rule.

Since curvature control between two intersecting isoparametric lines is independent, we are able to conduct cross curvature control without changing the feature curve. The cross curvatures at the starting and ending points are evaluated from the existing
Fig. 12. Modified automobile hood surface with planar feature curve. (a) Log-aesthetic planar curve with $\alpha = 0.5$, $\xi = 0.5$, $\theta_s = 0.0$ [rad] and $\theta_e = 1.2$ [rad], where the red portion of the curve is extracted for embedding. (b) Modified surface. (c) Curvature plot of the feature curve of (b). (d) Osculating circles of cross curvatures along the feature curve of (b). (e) Side view of (d). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

surface. We can modify these cross curvature values at both ends, say within 10% of the magnitude. From (32) we are able to find the corresponding tangent angles $\theta_s$ and $\theta_e$. Let us assume that we have $N$ spans along the feature curve, so the intermediate cross curvatures $\kappa_i^{cr}$ can be computed as follows:

$$\theta_i = \theta_s + \frac{\Delta \theta}{N} i, \quad i = 1, \ldots, N - 1,$$

$$\kappa_i^{cr} = \begin{cases} e^{-\Lambda \theta_i}, & \alpha = 1 \\ \frac{1}{((\alpha - 1) \Lambda \theta_i + 1)^{1/\alpha}}, & \text{otherwise} \end{cases}$$

(41)

(42)

Based on (42), we obtain the control points of the cross feature curves for each cross curvature $\kappa_i^{cr}$ using (9) to (19). The surface control points can be obtained by solving (26). In this example $\alpha = 1.0$ and $c = 0.0$ are used. Fig. 12(d) and (e) depicts the osculating circles of the cross curvatures along the feature curve where the radius of the circles vary log-aesthetically, whereas those of the original surface shown in Fig. 11(c) varies without any rule. Note that for the sake of clarity, the center of curvature is set opposite in the Figs. 11, 12, 14 and 15. Fig. 13(a) and (b) depicts the Gaussian curvature distribution of the surfaces of Figs. 11(a) and 12(b), respectively, which clearly show that the changes in the surface are limited only in the vicinity along the feature curve.

5.2.2. Space feature curve

The automobile hood B-spline surface consisting of $20 \times 10$ control points shown in Fig. 14(a) has symmetric space feature curves. Fig. 14(b) shows the curvature plot along the left feature curve and the osculating circles of the cross curvature along the right feature curve.

The user first determines the curve segment of the log-aesthetic curve by specifying the positions of the two endpoints, their tangents, the slopes $\alpha$ and $\beta$ of straight lines of the logarithmic curvature and torsion graphs, and the torsion parameter $\Omega$. Once these input parameters are determined, one can find the intrinsic equations from (36) and (37) for a type 1 curve and from (36) and (38)
Fig. 14. Automobile hood surface with two symmetric space feature curves. (a) Original surface. (b) Curvature plot along the left feature curve, and osculating circles of cross curvatures along the right feature curve of (a).

Fig. 15. Modified automobile hood surface of Fig. 14. (a) Log-aesthetic space curve projected on to the xy-plane. (b) Log-aesthetic space curve projected on to the yz-plane. (c) Log-aesthetic space curve is embedded into a automobile hood surface of Fig. 14. (d) Curvature plot along the left feature curve, and osculating circles of cross curvatures along the right feature curve of (c).

for a type 2 curve. Fig. 15(a) and (b) shows the log-aesthetic curve with $\alpha = 0.2$, $\beta = 0.7$, and $\Omega = 0.1$ of the type 1 curve. The shape of the space feature curve can be obtained by solving the system of first order differential equations (39) with initial values (44). The rest of the processes are the same as those for the planar feature curve. We note here that our algorithm cannot interpolate torsions at knots, and hence the resulting space curve does not reflect the torsions of the space log-aesthetic curve exactly at knots. In this example we let the cross curvature be constant along the feature curve.

Fig. 15(c) displays the embedded surface, while (d) shows the curvature plot along the left feature curve and the osculating circles of the cross curvature along the right feature curve of the embedded surface.

The curvature plot along the feature curve depicted in Fig. 15(d) shows clearly that the feature curve is a log-aesthetic curve, whereas that of the original surface (see Fig. 14(b)) varies without any rule. Moreover, Fig. 15(d) shows that the cross curvatures along the feature curve are constant through the constant radius of the osculating circles, whereas those of the original surface varies without any rule.

6. Summary and future work

The proposed method allows us to edit the curvatures of two isoparametric curves at the knots of existing uniform bi-cubic B-spline surfaces. The positions and the directions of unit tangents and unit normals at knots can also be edited interactively, or one can use those of the predefined values. Using these point-based techniques, we successfully embedded log-aesthetic curves into existing B-spline surfaces. Moreover, we were able to distribute the cross curvature with log-aesthetic variation along the isoparametric curves, where the cross curvatures could be maintained during the feature curve editing process.

The limitations of the proposed algorithm are as follows.

- The feature curve constraints could be imposed only on the isoparametric curve of the surface.
Although the log-aesthetic space curve is computed based on the intrinsic equations, our interpolation algorithm is not able to interpolate the torsion.

A large deformation of the feature curve from the original surface may induce unwanted bumps near the feature curves.

We provided an outline of the concept of our proposed algorithms and applied them to several cases. There are several possible extensions to the proposed algorithm, some of which are listed below.

We hope to combine advantages of our method and so-called fully free-form deformation features (δ-FP) [32,33] so that we are able to extend the feature curve constraints of the isoparametric curve to an arbitrary curve on a surface.

We plan to extend the current bi-cubic B-spline surfaces to higher degree surfaces.

Acknowledgments

This work is supported by the Japan Society for the Promotion of Science, Grants-in-Aid for Scientific Research under grant number 24360060. We would like to thank Ahmad Nasri for motivating this research, Abadulwahed Abbas and Shoichi Okaniwa for their constructive discussions at the early stage of the research, and Hiroshi Tadenuma for his assistance.

Appendix. Ternary subdivision of the uniform bi-cubic B-spline patch

We consider the left bottom \( \frac{1}{3} \) of the uniform bi-cubic B-spline patch \( 0 \leq u \leq \frac{1}{3}, 0 \leq v \leq \frac{1}{3} \) (see Fig. A.1) as follows:

\[
R_{ij}(u,v) = \begin{bmatrix}
1 & u & u^2 & u^3 \\
\frac{u}{3} & \frac{u^2}{9} & \frac{u^3}{27} & \\
\frac{u}{3} & \frac{u^2}{9} & \frac{u^3}{27} \\
v & v^2 & v^3
\end{bmatrix} [M][P]_{ij}[M]^T
\]

where

\[
[S] = \frac{1}{27} \begin{bmatrix}
10 & 16 & 1 & 0 \\
4 & 19 & 4 & 0 \\
1 & 16 & 10 & 0 \\
0 & 10 & 16 & 1
\end{bmatrix}.
\]

Let

\[
[R]_{kl} = \begin{bmatrix}
R_{kl} & R_{kl+3} & R_{kl+6} & R_{kl+9} \\
R_{kl+3} & R_{kl+6} & R_{kl+9} & R_{kl+12} \\
R_{kl+6} & R_{kl+9} & R_{kl+12} & R_{kl+15} \\
R_{kl+9} & R_{kl+12} & R_{kl+15} & R_{kl+18}
\end{bmatrix} = [S][P]_{ij}[S]^T.
\]

where \( k = 3i, l = 3j \), then (A.1) can be rewritten as

\[
r_{ij}(u,v) = \begin{bmatrix}
1 & u & u^2 & u^3
\end{bmatrix} [M][R]_{ij}[M]^T
\]

References


