Surface Faring Using Circular Highlight Lines

Yu Nishiyama[1], Yoh Nishimura[2], Takayuki Sasaki[3] and Takashi Maekawa[4]

[1]Yokohama National University, d06gb150@ynu.ac.jp
[2]Brother Industries, Ltd.
[3]Komatsu Ltd.
[4]Yokohama National University, maekawa@ynu.ac.jp

ABSTRACT

We herein propose a novel method for removing irregularities of B-spline surfaces via smoothing circular highlight lines. A circular highlight line is defined as a set of points on a surface such that the distance between a circular light source and an extended surface normal to be zero. Circular highlight lines allow us to capture the surface fairness in all directions, whereas conventional method, which uses a family of parallel straight lines for light sources, can capture the surface irregularity only in one direction. This method of correcting surface irregularities through circular highlight lines is intuitive and allows non-skilled persons to generate surfaces that can satisfy requirements imposed by downstream applications. Nonlinear equations that relate the difference between the circular highlight lines of the current surface and the target curves in the parameter space are formulated in terms of control points of the surface to be modified. The nonlinear governing equations are solved by Newton's method. The effectiveness of these algorithms is demonstrated through examples.

Keywords: circular highlight lines, surface fairing, surface interrogation, B-spline surface.

1. INTRODUCTION

Isophotes, reflection lines, and highlight lines are first-order interrogation methods that are used in the automotive industry to assess the fairness of a surface [2],[5],[9],[10],[12],[14]. Isophotes are curves of constant light intensity on a surface, created by a point light source located at infinity with the direction specified by the user. These curves can be used for detecting surface irregularities [12],[14]. If the surface is C^M continuous, then the isophote line will be C^{M-1} continuous. Reflection lines [10] simulate the mirror images of a family of radiating parallel straight lines on a smooth surface viewed from a fixed point. In this method, deviations of the surface from a smooth shape can be detected by irregularities of the reflection lines. These surface deviations are corrected by modifying the original surface so that the new surface has reflection lines without any irregularities. Beier and Chen [2] introduced a simplified reflection line model called highlight lines that are viewer-independent. In other words, in the computation of highlight lines, viewpoint location is not necessary, which is in contrast to the reflection lines. Recently, Yong et al. [16] developed an interactive method that generates highlight lines dynamically on a locally deforming NURBS surface that uses a Taylor expansion technique, instead of time consuming tracing processes.

Elber [6] introduced another interrogation tool called reflection ovals that consider the reflections of oval curves, which are not straight lines. Maekawa et al. [11] developed a novel method for surface interrogation called circular highlight lines and circular reflection lines. Unlike conventional first-order surface interrogation methods, such as reflection lines and highlight lines, in which a family of parallel straight lines is used for the light sources, circular highlight lines employ concentric circle light sources to evaluate the surface fairness. In this way, the surface fairness in all directions can be captured, whereas conventional methods can only capture the fairness in one direction in a single computation.

Chen et al. [4] presented a method for updating the control points of the NURBS surface to a desired shape automatically by specifying the shape of the highlight lines. A system of nonlinear equations is formulated to relate NURBS control points and reference points on the highlight line. However, this method has several limitations. First, the computational errors depend on the magnitude of the highlight line modification and the curvature of the surface, because the problem formulation relies on the linear approximation through a first-order Taylor expansion. Second, the method works well only for highlight lines that are predominantly oriented in the u and v directions, and not for diagonally oriented highlight lines or highlight lines with loops. In addition, practically, it is very difficult to predefine highlight lines in a 3D space that lie on an unknown surface.

Zhang and Cheng [17] studied a method for removing local irregularities of NURBS surface patches by modifying highlight lines for real-time interactive design. This method allows a designer to simply click a mouse button at the endpoints of the irregular portions of the highlight lines on the surface to make the surface fair. The nonlinear governing equations are linearized by predicting the positions and the first partial derivatives of the surface along the target highlight lines. Consequently, the magnitude of the highlight line movement will be small and, as a result, the method requires several user interactions to modify the surface.

In the present paper, instead of using conventional highlight lines, we employ circular highlight lines to remove irregularities of B-spline surfaces. As depicted in Fig. 1, in some cases, surface irregularities are more clearly detected by circular highlight lines than by conventional highlight lines. The proposed method follows the concept proposed by Zhang and Cheng [17] whereby the user interactively inputs the two endpoints of the irregular portions of each highlight line by clicking the mouse. However, in the present study, we set the target curves in the parameter space of the surface, instead of 3D space, which makes the formulation much simpler. Furthermore, we solve the nonlinear equations using Newton's method, allowing us to modify surfaces that have large irregularities with a small number of user interactions.

The present paper is organized as follows. Section 2 reviews the circular highlight line model. In Section 3, the governing equations along with the solution methodologies are introduced. Section 4 presents illustrative examples. Finally, Section 5 concludes the paper.

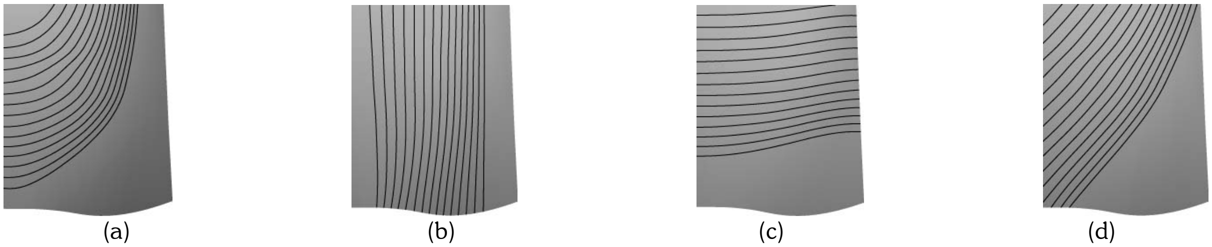


Fig. 1: (a) Surface with small local irregularities those are recognizable by circular highlight lines. In this figure,(b), (c), and (d) show that it is difficult to detect surface irregularities using conventional highlight lines.

2. CIRCULAR HIGHLIGHT LINE MODEL

The circular highlight line was introduced by Maekawa et al. [11] and is defined as a set of points on a surface such that the distance between a circular light source and an extended surface normal to the highlight lines is zero, as shown in Fig. 2(a). A parametric representation of the circular light source is given by

$$\mathbf{L}(\theta) = \mathbf{A} + R(\cos\theta\mathbf{n} + \sin\theta\mathbf{b}) \tag{2.1}$$

where $\mathbf{A}$ and R are the center point and the radius of the circular light source, respectively. The unit vectors $\mathbf{n}$ and $\mathbf{b}$ lie in a plane that contains the circular light and are orthogonal to each other. They form a frame (or trihedron) together with a unit vector $\mathbf{t}$, such that $\mathbf{t} = \mathbf{n} \times \mathbf{b}$, and hence $\mathbf{t}$ is perpendicular to the plane that contains the circular light. We also assume that the surface of interest is a parametric B-spline surface defined by a topologically rectangular set of control points $\mathbf{R}_{ij},\ 0 \le i \le M,\ 0 \le j \le N$ and two knot vectors $\mathbf{U} = (u_0, u_1, \cdots, u_{M+K})$ and $\mathbf{V} = (v_0, v_1, \cdots, v_{N+L})$ associated with each parameter *u* and *v*. The corresponding B-spline surface is given by

$$\mathbf{r}(u,v) = \sum_{i=0}^{M}\sum_{j=0}^{N}\mathbf{R}_{ij}N_{i,K}(u)N_{j,L}(v)\,, \tag{2.2}$$

where $N_{i,K}(u)$ and $N_{j,L}(v)$ are the *K-th* order and *L-th* order B-spline basis functions, respectively. We assume that the parameterization is well conditioned. Now let us define an extended surface normal vector $\mathbf{E}$ at a surface point $\mathbf{Q}$ as

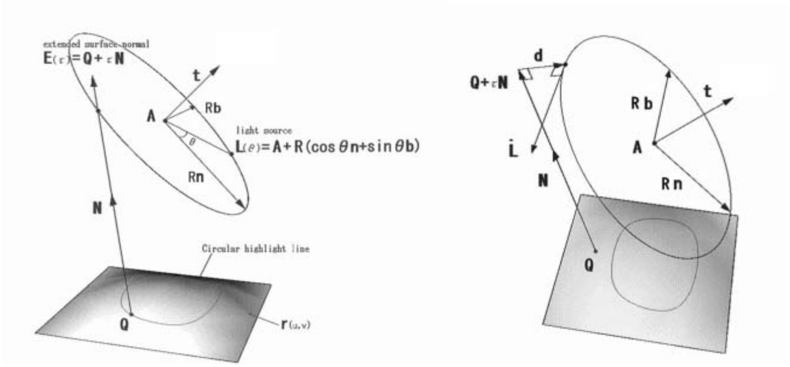
$$\mathbf{E}(\tau) = \mathbf{Q} + \tau\mathbf{N}\,, \tag{2.3}$$

where τ is a parameter and $\mathbf{N}$ is the unit surface normal vector at $\mathbf{Q}$

$$\mathbf{N}(u,v) = \frac{\mathbf{r}_u(u,v) \times \mathbf{r}_v(u,v)}{|\,\mathbf{r}_u(u,v) \times \mathbf{r}_v(u,v)\,|}\,. \tag{2.4}$$

As illustrated in Fig. 2 (b), the distance vector $\mathbf{d}$ directed from the line $\mathbf{E}(\tau)$ to the circle $\mathbf{L}(\theta)$ is given by

$$\mathbf{d} = \mathbf{A} + R(\cos\theta\mathbf{n} + \sin\theta\mathbf{b}) - (\mathbf{Q} + \tau\mathbf{N})\ . \tag{2.5}$$



(a) (b)

Fig. 2: (a) Definition of circular highlight line. (b) Definition of distance vector.

In addition, the squared distance function D is defined as

$$D(\tau,\theta) = \mathbf{d}\cdot\mathbf{d} = \Big|(\mathbf{A} \;+\; R(\cos\theta\mathbf{n} + \sin\theta\mathbf{b})) - (\mathbf{Q} + \tau\mathbf{N})\Big|^2 \quad . \tag{2.6}$$

To compute the minimum distance, we need to evaluate the stationary points of the squared distance function which satisfy the following two equations [10]:

$$\mathrm{D}_\tau(\tau,\theta)=\mathrm{D}_\theta(\tau,\theta)=0\,. \tag{2.7}$$

Using Eqn. (2.6), these two equations can be rewritten as follows:

$$\begin{aligned}(\mathbf{A}-\mathbf{Q})\cdot\mathbf{N} + R(\cos\theta\mathbf{n}\cdot\mathbf{N} - \sin\theta\mathbf{b}\cdot\mathbf{N}) &= \tau,\\ (\mathbf{A}-\mathbf{Q}-\tau\mathbf{N})\cdot(\cos\theta\mathbf{b} - \sin\theta\mathbf{n}) &= 0\ ,\end{aligned} \tag{2.8}$$

or in matrix form as

$$\begin{pmatrix} R\mathbf{n}\cdot\mathbf{N} & R\mathbf{b}\cdot\mathbf{N} \\ (\mathbf{A}-\mathbf{Q}-\tau\mathbf{N})\cdot\mathbf{b} & (\tau\mathbf{N}-\mathbf{A}+\mathbf{Q})\cdot\mathbf{n}\end{pmatrix}\begin{pmatrix}\cos\theta\\ \sin\theta\end{pmatrix} = \begin{pmatrix}\tau-(\mathbf{A}\text{ - }\mathbf{Q})\cdot\mathbf{N}\\ 0\end{pmatrix}. \tag{2.9}$$

Using Cramer's rule, we obtain

$$\cos\theta = \frac{(\tau-(\mathbf{A}\text{ - }\mathbf{Q})\cdot\mathbf{N})(\tau\mathbf{N}-\mathbf{A}+\mathbf{Q})\cdot\mathbf{n}}{Det},\quad \sin\theta = \frac{(\tau-(\mathbf{A}\text{ - }\mathbf{Q})\cdot\mathbf{N})(\tau\mathbf{N}-\mathbf{A}+\mathbf{Q})\cdot\mathbf{b}}{Det}\ , \tag{2.10}$$

where Det is the determinant of the matrix of Eqn. (2.9) given by,

$$Det = R(\mathbf{n}\cdot\mathbf{N})(\mathbf{Q}+\tau\mathbf{N}-\mathbf{A})\cdot\mathbf{n} + R(\mathbf{b}\cdot\mathbf{N})(\mathbf{Q}+\tau\mathbf{N}-\mathbf{A})\cdot\mathbf{b}\,. \tag{2.11}$$

If we denote

$$\mathbf{B} = \mathbf{A}-\mathbf{Q},\ \ \alpha = \mathbf{n}\cdot\mathbf{N},\ \ \beta = \mathbf{b}\cdot\mathbf{N},\ \ \gamma = \mathbf{B}\cdot\mathbf{n},\ \ \delta{=}\mathbf{B}\cdot\mathbf{b},\ \ \varepsilon = \mathbf{B}\cdot\mathbf{N}\,, \tag{2.12}$$

Eqn. (2.11) becomes

$$Det = R\alpha(\alpha\tau-\gamma) + R\beta(\beta\tau-\delta)\,. \tag{2.13}$$

Using the above notations Eqn. (2.12) and substituting Eqn. (2.10) into

$$\cos^2\ \theta + \sin^2\ \theta = 1\ \ , \tag{2.14}$$

we obtain a quartic equation in τ

$$c_4\tau^4{+}c_3\tau^3{+}c_2\tau^2 + c_1\tau^1 + c_0 = 0\ \ , \tag{2.15}$$

where

$$\begin{aligned} c_4 &= \alpha^2+\beta^2,\\ c_3 &= -2((\alpha\gamma+\beta\delta)+(\alpha^2+\beta^2)\varepsilon),\\ c_2 &= (\alpha^2+\beta^2)\varepsilon^2 + 4\varepsilon(\alpha\gamma+\beta\delta) + (\gamma^2+\delta^2) - R^2(\alpha^2+\beta^2)^2,\\ c_1 &= -2\Big((\alpha\gamma+\beta\delta)\varepsilon^2 + (\gamma^2+\delta^2)\varepsilon - R^2(\alpha^2+\beta^2)(\alpha\gamma+\beta\delta)\Big),\\ c_0 &= (\gamma^2+\delta^2)\varepsilon^2 - R^2(\alpha\gamma+\beta\delta)^2.\end{aligned} \tag{2.16}$$

An algorithm for solving the quartic equation is given by Hacke [7]. In summary, given a point $\mathbf{Q}$ on a parametric surface $\mathbf{r}(u,\ v)$, we can compute τ by solving the quartic equation, Eqn. (2.15), and then $\cos\theta$ and $\sin\theta$ from Eqn. (2.10), provided that Det is not zero. Finally, the distance vector $\mathbf{d}$ is obtained from Eqn. (2.5). There are cases when Det becomes zero, and these are discussed in [11]. The tangent vector of the circle is obtained by differentiating Eqn. (2.1) with respect to θ yielding

$$\dot{\mathbf{L}}(\theta) = R(-\sin\theta\mathbf{n} + \cos\theta\mathbf{b})\,. \tag{2.17}$$

Since, by definition, $\mathbf{d}\cdot\mathbf{N} = 0$ and $\mathbf{d}\cdot\dot{\mathbf{L}} = 0$ (see Fig. 2(b)), and since we have $(\mathbf{N}\times\dot{\mathbf{L}})\cdot\mathbf{N} = 0$ and $(\mathbf{N}\times\dot{\mathbf{L}})\cdot\mathbf{L} = 0$ from the definition of the triple scalar product, we can conclude that the distance vector $\mathbf{d}$ is parallel to $\mathbf{N}\times\dot{\mathbf{L}}$. A signed distance function d_s [2] can be defined by taking the dot product with the unit vector $\dfrac{\mathbf{N}\times\dot{\mathbf{L}}}{\left|\mathbf{N}\times\dot{\mathbf{L}}\right|}$ as follows:

$$d_s(u,v) = \left((\mathbf{A}\ +\ R(\cos\theta\mathbf{n} + \sin\theta\mathbf{b})) - (\mathbf{Q} + \tau\mathbf{N}(u,v))\right)\cdot\frac{\mathbf{N}(u,v)\times\dot{\mathbf{L}}(\theta)}{\left|\mathbf{N}(u,v)\times\dot{\mathbf{L}}(\theta)\right|}\ . \tag{2.18}$$

Circular highlight lines are points on a surface such that the signed distance function vanishes. If we construct a signed distance surface $(u,\ v,\ d_s(u,\ v))$ by evaluating the discrete values of d_s at the grid points in the uv parameter space, we can easily compute the pre-image of the circular highlight lines using the contour algorithm [3], where curves of zero height are computed through surface-plane intersection problems. Although the accuracy of the contouring algorithm depends on the choice of grid resolution, we employed the contouring algorithm for its simplicity and speed. A more realistic light source can be realized by replacing the circle with a torus of given radius ρ , which is obtained by simply finding the points satisfying $d_s = \rho$ and $d_s = -\rho$. These two curves are called circular highlight band boundary curves and form a circular highlight band [2],[11].

3. FORMULATION OF GOVERNING EQUATIONS

3.1 Basic Concept

The fairing process starts with finding local irregularities of the surface by observing the behavior of the circular highlight lines. While Zhang and Cheng [17] determined the target curve in 3*D* space, we set the target curve in the parameter space. This is possible, as we assuming that the parameterization is well conditioned. The user first inputs interactively the two endpoints of the irregular portions of the pre-images of each circular highlight line by clicking the mouse, as shown in Fig. 3. Then, we generate a smooth target curve that interpolates the positions and the slopes of the two endpoints using a cubic Hermite curve (see Fig. 3(d)). The control points of the surface to be modified will then be selected automatically based on the locations of the endpoints. Finally, we modify the control points of the surface by solving a set of nonlinear equations using Newton's method.

3.2 Cubic Hermite Interpolation

The target curve is determined by interpolating the two endpoints and their tangent vectors of the irregular portion for each highlight line in the parameter space, as shown in Fig. 4(a). Let us denote the two endpoints and their tangent vectors of the *i-th* pre-image of the circular highlight line by $\mathbf{e}_i^1$, $\mathbf{e}_i^2$, $\mathbf{t}_i^1$ and $\mathbf{t}_i^2$. The corresponding cubic Hermite curve $\mathbf{h}_i(t)$ can then be obtained as follows:

$$\mathbf{h}_i(t) = H_0(t)\mathbf{e}_i^1 + H_1(t)\mathbf{t}_i^1 + H_2(t)\mathbf{e}_i^2 + H_3(t)\mathbf{t}_i^2,\ \ i = 1,2,\cdots,l, \tag{3.1}$$

where

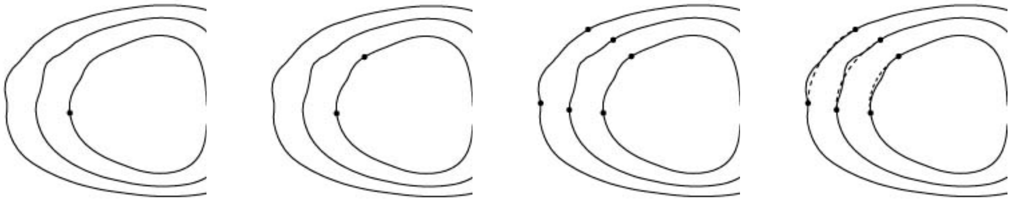
$$H_0(t) = 2t^3 - 3t^2 + 1,\ \ H_1(t) = -2t^3 + 3t^2,\ H_2(t) = t^3 - 2t^2 + t,\ \ H_3(t) = t^3 - t^2, \tag{3.2}$$

are cubic Hermite polynomials. The magnitudes of the two tangent vectors are determined empirically as follows:

$$\left\|\mathbf{t}_i^1\right\| = \left\|\mathbf{t}_i^2\right\| = \left\|\mathbf{e}_i^1 - \mathbf{e}_i^2\right\|, \tag{3.3}$$

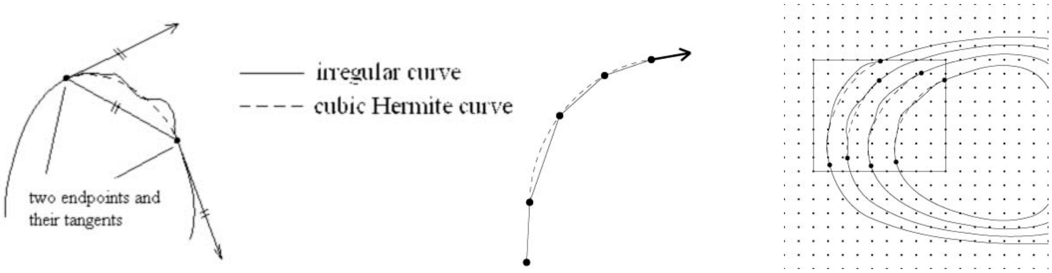
which is depicted in Fig. 4(a). The directions of the tangent vectors at the two endpoints are estimated by taking the first derivatives of the cubic Bézier curves interpolating the four points of the regular portions of the pre-images of the circular highlight lines (see Fig. 4(b)).

Once all of the endpoints are selected by the user, a bounding box consisting of knots that encloses these endpoints in the parameter space is generated, as shown in Fig. 5. All of the control points that affect this bounding box will be selected as the control points to be modified.



(a) (b) (c) (d)

Fig. 3: (a) The user selects one of the endpoints of the irregular portion of the *i-th* pre-image of the circular highlight line. (b) The user selects the other endpoint of the irregular portion of the *i-th* pre-image of circular highlight line. (c) Two endpoints of all of the pre-images of the circular highlight lines are selected. (d) Target curves (dashed lines) are computed using cubic Hermite interpolation.



(a) (b) (c)

Fig. 4: Cubic Hermite interpolation. (a) Magnitude of the tangent vector. (b) Direction of the tangent vector.(c) Bounding box that encloses the irregular region (grid points represent knots).

3.3 Solution Methodology

We take $n(i)$ sampling points from each interpolated cubic Hermite curve as follows:

$$\mathbf{h}_i(t_j) = (u_{ij}, v_{ij}),\ t_j = \frac{j - 1}{n(i) - 1},\ i = 1,2,\cdots,l,\ j = 1,2,\cdots,n(i), \tag{3.4}$$

Where $n(i)$ is given by

$$n(i) = 2 + \left[\frac{\left\| \mathbf{e}_i^1 - \mathbf{e}_i^2 \right\|}{s}\right]. \tag{3.5}$$

The bracket $[x]$ in Eqn. (3.5) is the Gauss symbol, and provides the greatest integer that is less than or equal to x, and s is a constant. In our implementation, we used $s = 50$.

Let $\mathbf{R} = (r_1, r_2 \cdots, r_m)$ be the selected components of the control points of the input B-spline surface that will be modified through the fairing process. Now let us assume that signed distance function $d_s(u_{ij}, v_{ij})$ (Eqn. (2.18)) at the *j-th* sampling point of the *i-th* cubic Hermite curve is related by $\mathbf{R}$ through a nonlinear function $F_{ij}(r_1,\ r_2,\ .\ .\ .\ ,\ r_m)$ as follows:

$$F_{ij}(r_1,\ r_2,\ .\ .\ .\ ,\ r_m) = d_s(u_{ij}, v_{ij}),\quad i = 1,2,\cdots,l,\quad j = 1,2,\cdots,n(i). \tag{3.6}$$

This system of nonlinear equations with unknowns $r_1,\ r_2,\cdots\ ,r_m$ can be solved by Newton's method. At each Newton iteration, we want to find small displacements of the selected components of the control points of the input B-spline surface $\Delta\mathbf{r} = (\Delta r_1,\ \Delta r_2, \cdots\ ,\ \Delta r_m)$, such that

$$F_{ij}(r_1 + \Delta r_1,\ r_2 + \Delta r_2,\ .\ .\ .\ ,\ r_m + \Delta r_m) = 0,\quad i = 1,2,\cdots,l,\quad j = 1,2,\cdots,n(i). \tag{3.7}$$

In other words, we move the control points of the surface so that the signed distance functions $d_s(u_{ij}, v_{ij})$ at (u_{ij}, v_{ij}) vanish for $i = 1,2,\cdots,l$ and $j = 1,2,\cdots,n(i)$, which means that these points are the pre-images of the circular highlight lines. Now, if we apply a Taylor expansion to Eqn. (3.7) and neglect the higher-order terms, which are sufficiently small, we have

$$\begin{aligned} & F_{ij}(r_1 + \Delta r_1,\ r_2 + \Delta r_2,\ .\ .\ .\ ,\ r_m + \Delta r_m) \\ &= F_{ij}(r_1,\ r_2,\ .\ .\ .\ ,\ r_m) + \frac{\partial F_{ij}}{\partial r_1}\Delta r_1 + \frac{\partial F_{ij}}{\partial r_2}\Delta r_2 + \cdots + \frac{\partial F_{ij}}{\partial r_m}\Delta r_m \\ &= d_s(u_{ij}, v_{ij}) + \frac{\partial F_{ij}}{\partial r_1}\Delta r_1 + \frac{\partial F_{ij}}{\partial r_2}\Delta r_2 + \cdots + \frac{\partial F_{ij}}{\partial r_m}\Delta r_m \\ &= 0, \end{aligned} \qquad (3.8)$$

for $i = 1,2,\cdots,l$ and $j = 1,2,\cdots,n(i)$, which results in a matrix:

$$\begin{pmatrix} \frac{\partial F_{11}}{\partial r_1} & \frac{\partial F_{11}}{\partial r_2} & \cdots & \frac{\partial F_{11}}{\partial r_m} \\ \vdots & \vdots & & \vdots \\ \frac{\partial F_{l\,\mathrm{n}(1)}}{\partial r_1} & \frac{\partial F_{l\,\mathrm{n}(1)}}{\partial r_2} & \cdots & \frac{\partial F_{l\,\mathrm{n}(1)}}{\partial r_m} \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ \frac{\partial F_{l\,\mathrm{n}(l)}}{\partial r_1} & \frac{\partial F_{l\,\mathrm{n}(l)}}{\partial r_2} & \cdots & \frac{\partial F_{l\,\mathrm{n}(l)}}{\partial r_m} \end{pmatrix} \begin{pmatrix} \Delta r_1 \\ \Delta r_2 \\ \vdots \\ \Delta r_m \end{pmatrix} = - \begin{pmatrix} d_s(u_{11}, v_{11}) \\ \vdots \\ d_s(u_{ln(1)}, v_{ln(1)}) \\ \vdots \\ \vdots \\ d_s(u_{l\,\mathrm{n}(l)}, v_{l\,\mathrm{n}(l)}) \end{pmatrix}. \qquad (3.9)$$

The elements of the Jacobian matrix are evaluated numerically. For example, the elements of the *k-th* column of the Jacobian matrix can be evaluated as follows:

1. Move only r_k by small amount of Δw . paper.
2. Compute functions $\mathrm{F_{ij}(r_1,\ .\ .\ .,r_k,\ \cdots,r_m)}$ and $\mathrm{F_{ij}(r_1,\ .\ .\ .,r_k + \Delta w,\ \cdots,r_m)}$ at (u_{ij}, v_{ij}) for $i = 1,2,\cdots,l$ and $j = 1,2,\cdots,n(i)$.
3. The partial derivatives are computed as follows:

$$\begin{aligned} \frac{\partial F_{11}}{\partial r_k} &= \frac{F_{11}(r_1,\cdots,\ r_k + \Delta w,\ \cdots,\ r_m)\ \text{-}\ F_{ij}(r_1,\ r_2,\cdots,\ r_m)}{\Delta w} \\ &\vdots \\ \frac{\partial F_{1n(1)}}{\partial r_k} &= \frac{F_{1n(1)}(r_1,\cdots,\ r_k + \Delta w,\ \cdots,\ r_m)\ \text{-}\ F_{1n(1)}(r_1,\ r_2,\cdots,\ r_m)}{\Delta w} \\ &\vdots \\ \frac{\partial F_{ln(l)}}{\partial r_k} &= \frac{F_{ln(l)}(r_1,\cdots,\ r_k + \Delta w,\ \cdots,\ r_m)\ \text{-}\ F_{ln(l)}(r_1,\ r_2,,\ r_m)}{\Delta w} \end{aligned} \qquad (3.10)$$

We repeat steps 1~3 for $k = 1,2,\cdots,m.$ We used $\Delta w = 10^{-4}$ for all of the computations in the present.

The size of the Jacobian matrix in Eqn. (3.9) is $\left(\sum_{i=1}^{l} n(i)\right) \times m$, where *l* is the number of selected circular highlight lines, $n(i)$ is the number of sampling points on the i-*th* cubic Hermite curve, and *m* is the number of components of the selected control points of the input B-spline surface that is to be modified through the fairing process. We can select l and $n(i)$, such that the $\left(\sum_{i=1}^{l} n(i)\right) \times m$ linear system is always an overdetermined system. The overdetermined linear system can be solved in a least squares sense. We employed the Singular Value Decomposition (SVD) [1] for solving the linear system. In our implementation, we set the singular values to zero if they are smaller than $w_{\max} \times 0.05$, where $w_{\max}$ is the maximum singular value of the matrix. The correction vector at each iteration is added to the current solution vector as follows:

$$\mathbf{R}^{(p+1)} = \mathbf{R}^{(p)} + \Delta\mathbf{r}\ , \qquad (3.11)$$

where the superscripts (p+1) and (p) denote the p+1-*th* and p-*th* iterations, respectively. Since the corrections are based on a first-order Taylor approximation, the regular Newton's method may not be sufficient for a very complex surface. If the vector norm of the correction vector is large, this is an indication that the problem is highly nonlinear and may produce a divergent iteration. Therefore, we assume that the iteration is converging if

$$| d_s |^{(p+1)}_{\max} < | d_s |^{(p)}_{\max} \text{, and} | d_s |^{(p+1)}_{avg} < | d_s |^{(p)}_{avg} \text{,} \tag{3.12}$$

where subscripts *max* and *avg* denote the maximum and average magnitudes of the signed distance function, respectively. If this condition is violated, we employ a step correction procedure:

$$\mathbf{R}^{(p+1)} \ = \ \mathbf{R}^{(p)} \ + \ \mu\Delta\mathbf{r}\,, \tag{3.13}$$

where $0 < \mu \le 1$. If $\mu = 1$, then the equation reduces to the regular Newton's method Eqn(3.11), whereas if $\mu < 1$, the rate of convergence will be less than quadratic. We used $\mu = 0.1$ in our implementation. We repeat these steps until

$$\frac{| d_s |^{(p+1)}_{avg} - | d_s |^{(p)}_{avg}}{\left| d_s \right|^{(p)}_{avg}} < \varepsilon \,\text{.} \tag{3.14}$$

4. EXAMPLES

In this section, we demonstrate how the proposed algorithms work for fairing B-spline surfaces. The surfaces in the examples include a surface with mainly elliptic points (positive Gaussian curvature), a surface with mainly hyperbolic points (negative Gaussian curvature), and a real engineering surface, i.e., an automotive roof panel. All of these examples were executed on a PC having a Pentium® IV 2.4 GHz processor and 512 MB of memory. We set the tolerance ε for the Newton's method to $\varepsilon = 10^{-4}$ in all examples.

The first example, shown in Fig. 5, is a dome-like bi-cubic B-spline surface with 15×15 control points, defined on clamped uniform knots, consisting mainly of elliptic points. Fig. 5(a) shows the B-spline surface with circular highlight lines, and Fig. 5(b) shows the pre-image of the circular highlight lines with endpoints bounding the local irregularities. Fig. 5(c), 5(d), 5(e), and 5(f) show the faired surface and its pre-images of the circular highlight lines after the first and second fairings. Tab. 1 lists all of the computational results, including the number of fairing processes, the number of light sources used in the computation, the number of iterations in Newton's method, the size of the Jacobian matrix, and the computational time in seconds. The computation time for user interactions, i.e., for selecting endpoints, is not included in all of the examples.

The second example is a saddle-like surface consisting of mainly hyperbolic points, as shown in Fig 6, and is expressed by a bi-cubic B-spline surface with 21×20 control points defined on clamped uniform knots. The computational results are presented in Tab. 1.

The third example is an automotive roof panel expressed by a bi-cubic B-spline surface with 21×22 control points defined on clamped uniform knots (see Fig. 7).

Model	Faring	# of light sources	# of iterations	Size of matrix	Time(sec)
Dome-like surface	1st fairing	12	5	192×36	0.95
	2nd fairing	8	8	140×30	1.12
Saddle-like surface	1st fairing	11	4	460×165	6.32
	2nd fairing	5	8	120×54	1.48
	3rd fairing	7	7	94×27	0.47
Automotive roof panel	1st fairing	18	6	273×108	5.68
	2nd fairing	15	4	202×72	1.56
	3rd fairing	19	5	249×60	2.21
	4th fairing	9	7	143×24	0.77
	5th fairing	16	4	273×60	1.50

Tab. 1: Computational results of three examples.

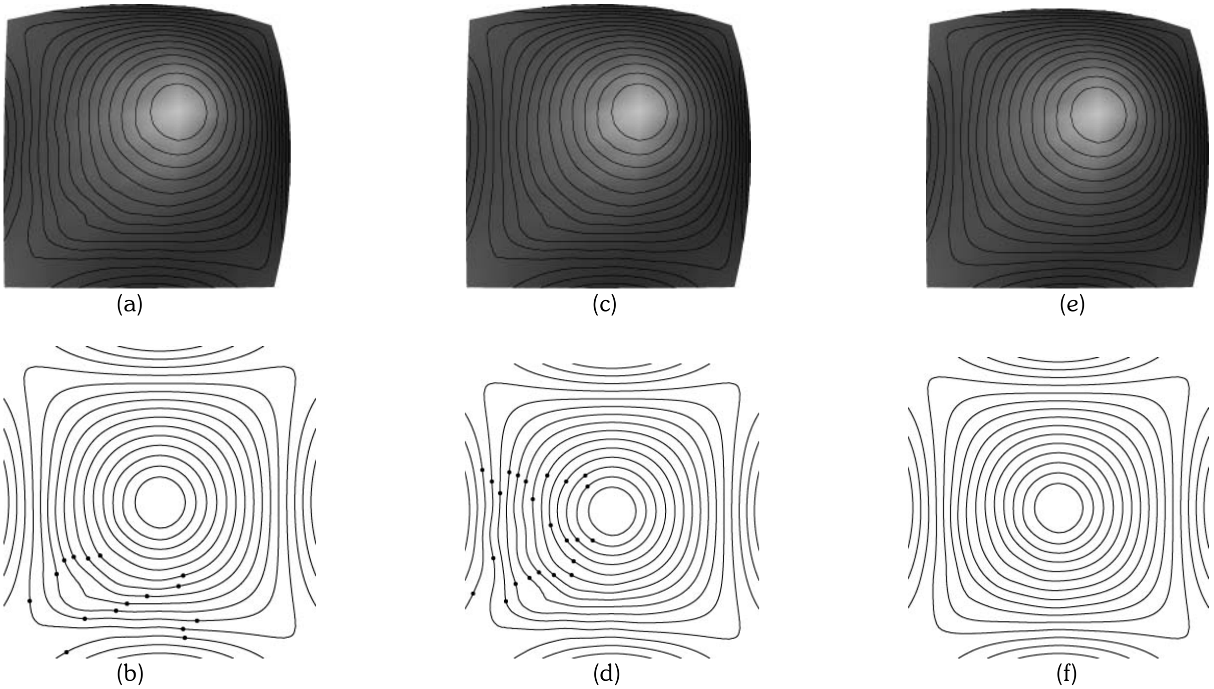


Fig. 5: (a) Circular highlight lines on a bi-cubic B-spline surface, consisting of mainly elliptic points, with local irregularities. (b) Pre-image of the circular highlight lines of surface (a) with endpoints bounding the local irregularities. (c) Faired surface after one fairing process. (d) Pre-image of the circular highlight lines of surface (c). (*e*) Faired surface after 2*nd* fairing process. (f) Pre-image of the circular highlight line of surface (*e*).

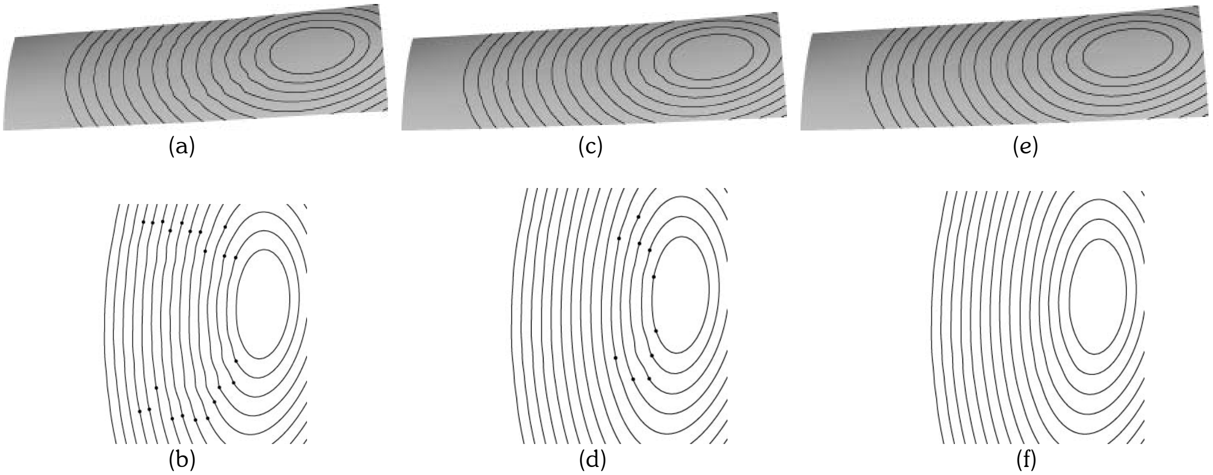


Fig. 6: (a) Circular highlight lines on a saddle-like bi-cubic B-spline surface, consisting of mainly hyperbolic points, with local irregularities. (b) Pre-image of the circular highlight lines of surface (a) with endpoints bounding the local irregularities. (c) Faired surface after one fairing process. (d) Pre-image of circular highlight lines of surface (c). (*e*) Faired surface after 3*rd* fairing process. (f) Pre-image of circular highlight lines of surface (*e*).

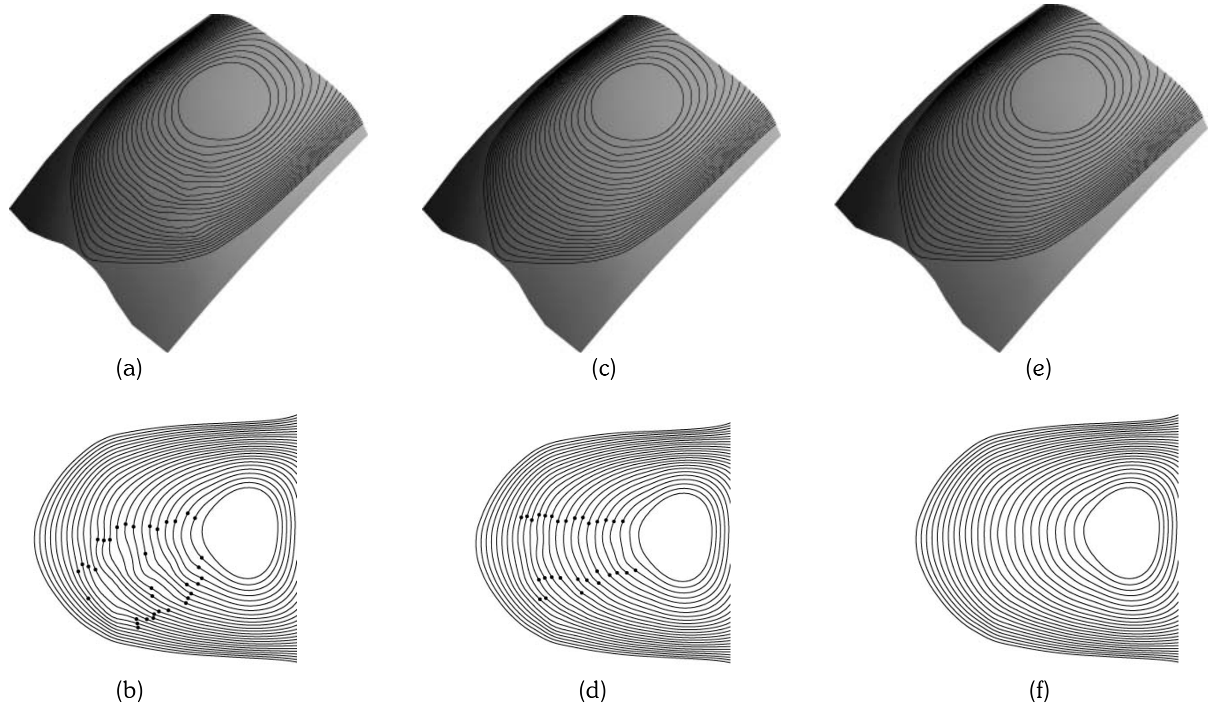


Fig. 7: (a) Automotive roof panel with local irregularities, which is expressed by a bi-cubic B-spline surface consisting of 462 control points. (b) Pre-image of the circular highlight lines of surface (a) with endpoints bounding the local irregularities. (c) Faired surface after the first fairing process. (d) Pre-image of the circular highlight lines of surface (c). (*e*) Faired surface after the 5*th* fairing process. (f) Pre-image of the circular highlight lines of surface (*e*).

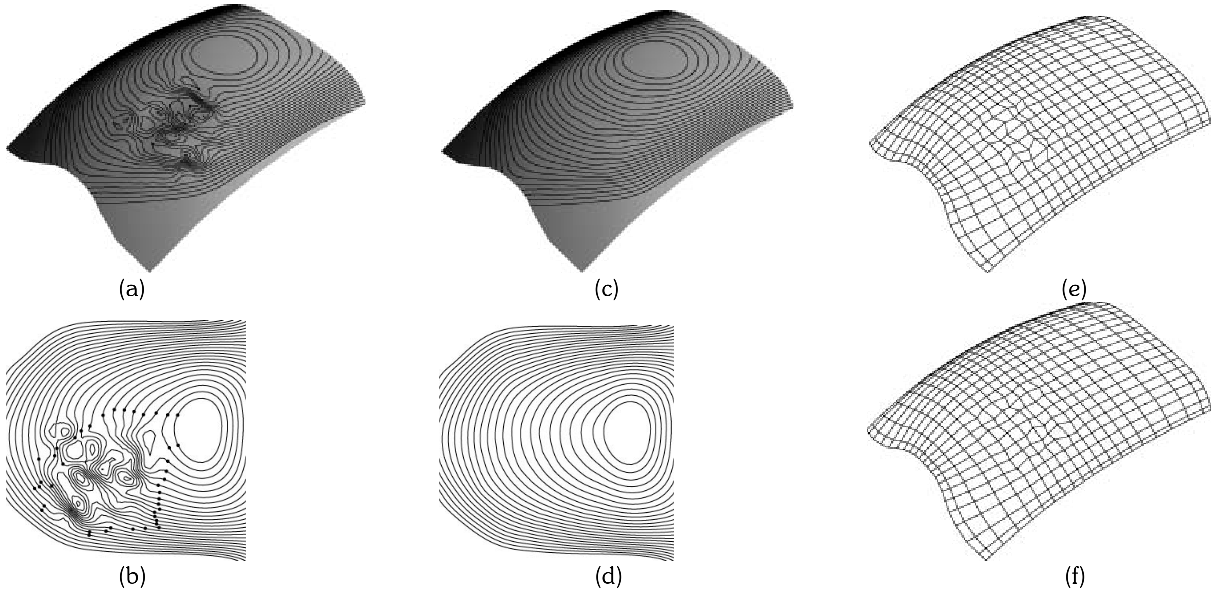


Fig. 8: (a) Surface with large irregularities. (b) Pre-images of circular highlight lines of surface (a). (c) Surface after the 10th fairing. (d) Pre-images of the circular highlight lines of surface (c). (*e*) Control polygon of surface (a). (f) Control polygon of the surface after the 10*th* fairing.

The proposed algorithm has some limitations. If the surface irregularities are located on the boundary of the surface, we may not be able to provide good target lines. For the case in which the surface irregularities are too large, as shown in Fig. 8(a) and 8(b), even if we are able to remove the irregularities of the pre-images of the circular highlight lines, as depicted in (d), the irregularities of the surface are not necessarily removed, even after the 10*th* fairing, as shown in 8(c). Fig. 8(e) and 8(f) show the control nets of the B-spline of the surface before and after the fairing, respectively**.**

5. CONCLUSIONS

We have introduced a novel interactive method to modify irregularities of B-spline surfaces via smoothing of circular highlight lines. The newly developed method is intuitive and allows non-skilled persons to generate surfaces that can satisfy the requirements imposed by the downstream applications. The method is demonstrated through several examples to be robust and effective for real-time interactive design.

6. REFERENCES

[1] Anderson, E. et al.: LAPACK Users' Guide, SIAM, Philadelphia, 1999.

[2] Beier, K.–P.; Chen, Y.: Highlight–line algorithm for realtime surface-quality assessment, Computer-Aided Design, 26(4), 1994, 268–277.

[3] Bourke, P.: CONREC: A Contouring Subroutine, http://astronomy.swin.edu.au/pbourke/projection/conrec/, July 1987.

[4] Chen, Y.; Beier, K.-P.; Papageorgiou, D.: Direct highlight line modification on NURBS surfaces, Computer Aided Geometric Design, 14(6), 1997, 583–601.

[5] Choi, I.; Lee, K.: Efficient generation of reflection lines to evaluate car body surfaces, Mathematical Engineering in Industry, 7(2), 1998, 233–250.

[6] Elber, G.: Curve evaluation and interrogation on surfaces, Graphical Models, 63(3): 2001,197–210.

[7] Hacke, J. E.: A simple solution of the general quartic, American Mathematical Monthly, 48(5), 1941, 327–328.

[8] Hagen, H.; Hahmann, S.; Schreiber, T.; Nakajima, Y.; Wördenweber, B.; Hollemann-Grundstedt, P.: Surface interrogation algorithms, IEEE Computer Graphics and Applications, 12(5), 1992, 53–60.

[9] Hoshcek, J; Kaklis, P. D.: Advanced Course on FAIRSHAPE. B.G. Teubner, Stuttart, 1996

[10] Klass, R.: Correction of local surface irregularities using reflection lines, Computer-Aided Design, 12(2), 73–76, 1980.

[11] Maekawa, T.; Nishimura, Y.; Sasaki, T.: Circular Highlight/Reflection Lines, Computer-Aided Design & Applications, 2(1-4), 2005, 291-300.

[12] Patrikalakis, N. M.; Maekawa, T.: Shape Interrogation for Computer Aided Design and Manufacturing Heidelberg, Germany, Springer-Verlag, 2002.

[13] Piegl, L. A.; Tiller. W.: The NURBS Book, Springer, New York, 1995.

[14] Poeschl, T.: Detecting surface irregularities using isophotes, Computer Aided Geometric Design, 1(2), 1984, 163–168.

[15] Struik, D. J.: Lectures on Classical Differential Geometry, Addison-Wesley, Cambridge, MA, 1950.

[16] Yong, J. -H.; Cheng, F.; Chen, Y.; Stewart, P.; Miura, K. T.: Dynamic highlight line generation for locally deforming NURBS surfaces, Computer-Aided Design, 35(10), 2003, 881–892.

[17] Zhang, C.; Cheng, F.: Removing local irregularities of NURBS surfaces by modifying highlight lines, Computer-Aided Design, 30(12):923–930, 1998.