Circular Highlight/Reflection Lines

Takashi Maekawa, Yoh Nishimura and Takayuki Sasaki

Yokohama National University, maekawa@ynu.ac.jp

ABSTRACT

We present a novel method to assess the fairness of class A surfaces. The fairness of class A surfaces are often assessed by reflection curves on the surfaces of a family of parallel straight fluorescent lights, namely reflection lines and highlight lines. Unlike reflection lines and highlight lines, where a family of parallel straight lines are used for the light sources, we propose to use concentric circles as light sources. In this way we can capture the surface fairness in all directions, whereas the conventional methods can capture the fairness only in one direction. Illustrative examples show substantial merits of this method over the conventional straight line light based surface interrogation methods.

Keywords: class A surface, highlight lines, reflection lines, isophotes, surface fairing, surface interrogation, NURBS surface.

1. INTRODUCTION

Free-form surfaces are used in the bodies of ships, automobiles and aircraft, which have both functionality and attractive shape requirements. Outside parts of many electronic devices as well as consumer products are also designed with aesthetic shapes, which involve free-form surfaces. These surfaces are called class A surfaces, and various interrogation methods have been developed to assess the fairness of these class A surfaces.

Isophotes, *reflection lines*, and *highlight lines* are the first-order interrogation methods that are used in the automotive industry to assess the fairness of a surface [1. 6, 8, 11, 14]. *Isophotes* are curves of constant light intensity on a surface, created by a point light source at infinity with direction specified by the user. These curves can be used for the detection of surface irregularities [8, 14]. If the surface is C^M continuous then the isophote line will be C^{M-1} continuous. *Reflection lines* [8] 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. Choi and Lee [4] applied the Blinn-Newell type of reflection mapping, which uses simple and physically acceptable mapping algorithm, to generate reflection lines on a trimmed NURBS surface. Choi and Lee [4] also provide a thorough review of this topic. Kanai [7] computed reflection lines of a chain of small circular light sources along straight lines to capture the surface fairness in all directions.

Beier and Chen [1] introduced a simplified reflection line model called *highlight lines*, which is merely an orthogonal projection of a straight line onto a surface [12, 15], and hence it is viewer-independent. In other words, in the computation of highlight lines viewpoint location is not necessary in contrast to the reflection lines. Chen et al. [3] presented a method to update the control points of the NURBS surface [13] to a desired shape automatically via specifying the shape of highlight lines. Zhang and Cheng [18] studied a method to remove local irregularities of NURBS surface patches by modifying its highlight lines for real time interactive design. Recently, Yong et al. [16] developed an interactive method that generates highlight lines dynamically on a locally deforming NURBS surface using a Taylor expansion technique instead of time consuming tracing processes.

In this paper we extend the concept of highlight/reflection lines to *circular highlight/reflection lines* [10] by replacing a family of parallel light lines to concentric light circles so that we can capture the surface fairness in all directions. The paper is organized as follows: Section 2 introduces the circular highlight line method. Contouring methods for determining the pre-images of the circular highlight lines are given in Section 3. In Section 4 complexity analysis of the circular highlight lines are given. In Section 5 we show that the circular highlight line algorithm can be easily

applied to the computation of circular reflection lines. Illustrative examples are presented in Section 6. Finally Section 7 concludes the paper.

2. CIRCULAR HIGHLIGHT LINE MODEL

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 at the highlight lines is zero as shown in Fig. 1 (a). A parametric representation of the circular light source is given by

$$\mathbf{L}(\theta) = \mathbf{A} + R(\cos\theta\mathbf{n} + \sin\theta\mathbf{b}), \tag{1}$$

where $\mathbf{A}$ and R is the center point and the radius of the circular light source, respectively. The unit vectors $\mathbf{n}$ and $\mathbf{b}$ lie in the plane that contains the circular light and are orthogonal. 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 NURBS surface $\mathbf{r}(u, v)$ where $0 \le u, v \le 1$. Now let us define an extended surface normal vector $\mathbf{E}$ at a surface point $\mathbf{Q}$ as

$$\mathbf{E}(\tau) = \mathbf{Q} + \tau\mathbf{N}, \tag{2}$$

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

$$\mathbf{N}(u,v) = \frac{\mathbf{r}_u(u,v) \times \mathbf{r}_v(u,v)}{\left|\mathbf{r}_u(u,v) \times \mathbf{r}_v(u,v)\right|}. \tag{3}$$

As illustrated in Fig. 1 (b), distance vector $\mathbf{d}$ directed from the line $\mathbf{E}(\tau)$ to the circle $\mathbf{L}(\theta)$ is given by

$$\mathbf{d} = \mathbf{A} + R(\cos\theta\mathbf{n} + \sin\theta\mathbf{b}) - (\mathbf{Q} + \tau\mathbf{N}). \tag{4}$$

Also the squared distance function D is defined as

$$D(\tau,\theta) = \mathbf{d} \cdot \mathbf{d} = \left|(\mathbf{A} + R(\cos\theta\mathbf{n} + \sin\theta\mathbf{b})) - (\mathbf{Q} + \tau\mathbf{N})\right|^2. \tag{5}$$

To compute the minimum distance, we need to evaluate the stationary points of the squared distance function, which satisfy the following two equations [11]

$$D_\tau(\tau,\theta) = D_\theta(\tau,\theta) = 0. \tag{6}$$

Using (5), these two equations can be rewritten as follows:

$$(\mathbf{A} - \mathbf{Q}) \cdot \mathbf{N} + R(\cos\theta\mathbf{n} \cdot \mathbf{N} + \sin\theta\mathbf{b} \cdot \mathbf{N}) = \tau, \tag{7}$$

$$(\mathbf{A} - \mathbf{Q} - \tau\mathbf{N}) \cdot (\cos\theta\mathbf{b} - \sin\theta\mathbf{n}) = 0, \tag{8}$$

or in a matrix form

$$\begin{pmatrix} R\mathbf{n} \cdot \mathbf{N} & R\mathbf{b} \cdot \mathbf{N} \\ (\mathbf{A} - \mathbf{Q} - \tau\mathbf{N}) \cdot \mathbf{b} & (\tau\mathbf{N} - \mathbf{A} + \mathbf{Q}) \cdot \mathbf{n} \end{pmatrix} \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} = \begin{pmatrix} \tau - (\mathbf{A} - \mathbf{Q}) \cdot \mathbf{N} \\ 0 \end{pmatrix}. \tag{9}$$

Using Cramer's rule, we obtain

$$\cos\theta = \frac{(\tau - (\mathbf{A} - \mathbf{Q}) \cdot \mathbf{N})(\tau\mathbf{N} - \mathbf{A} + \mathbf{Q}) \cdot \mathbf{n}}{Det}, \quad \sin\theta = \frac{(\tau - (\mathbf{A} - \mathbf{Q}) \cdot \mathbf{N})(\tau\mathbf{N} - \mathbf{A} + \mathbf{Q}) \cdot \mathbf{b}}{Det}, \tag{10}$$

where *Det* is the determinant of the matrix of (9) given by

$$Det = R(\mathbf{n} \cdot \mathbf{N})(\mathbf{Q} + \tau\mathbf{N} - \mathbf{A}) \cdot \mathbf{n} + R(\mathbf{b} \cdot \mathbf{N})(\mathbf{Q} + \tau\mathbf{N} - \mathbf{A}) \cdot \mathbf{b}. \tag{11}$$

If we denote

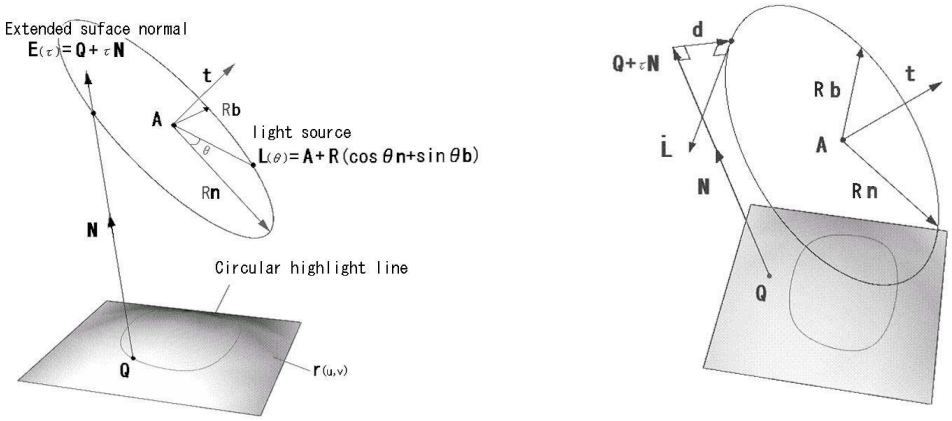
$$\mathbf{B} = \mathbf{A} - \mathbf{Q},\ \alpha = \mathbf{n} \cdot \mathbf{N},\ \beta = \mathbf{b} \cdot \mathbf{N},\ \gamma = \mathbf{B} \cdot \mathbf{n},\ \delta = \mathbf{B} \cdot \mathbf{b},\ \varepsilon = \mathbf{B} \cdot \mathbf{N}, \tag{12}$$

Equation (11) becomes

$$Det = R\alpha(\alpha\tau - \gamma) + R\beta(\beta\tau - \delta). \tag{13}$$

Using the above notations (12) and substituting Equations (10) into

$$\cos^2\theta + \sin^2\theta = 1, \tag{14}$$



(a) (b)

Fig. 1. (a) Definition of circular highlight line. (b) Definition of distance vector

we obtain a quartic equation in τ

$$c_4\tau^4 + c_3\tau^3 + c_2\tau^2 + c_1\tau + c_0 = 0\,, \tag{15}$$

where

$$c_4 = \alpha^2 + \beta^2 \tag{16}$$

$$c_3 = -2((\alpha\gamma + \beta\delta) + (\alpha^2 + \beta^2)\varepsilon) \tag{17}$$

$$c_2 = (\alpha^2 + \beta^2)\varepsilon^2 + 4\varepsilon(\alpha\gamma + \beta\delta) + (\gamma^2 + \delta^2) - R^2(\alpha^2 + \beta^2)^2 \tag{18}$$

$$c_1 = -2((\alpha\gamma + \beta\delta)\varepsilon^2 + (\gamma^2 + \delta^2)\varepsilon - R^2(\alpha^2 + \beta^2)(\alpha\gamma + \beta\delta)) \tag{19}$$

$$c_0 = (\gamma^2 + \delta^2)\varepsilon^2 - R^2(\alpha\gamma + \beta\delta)^2\,. \tag{20}$$

Using line geometry [15], it can be also shown that the degree of the problem is quartic. The roots of the fourth order polynomial equation was first solved by Ferrari (1522-1565) [17], a student of Cardano's. A source code for solving quartic equation, based on a method by Hacke [5] (see Appendix A), is available through the internet [9]. In summary, given a point $\mathbf{Q}(u,v)$ on a parametric surface $\mathbf{r}(u,v)$, we can compute τ by solving the quartic equation (15) and then $\cos\theta$ and $\sin\theta$ from (10), provided that *Det* is not zero, and finally the distance vector $\mathbf{d}$ from (4). Figures 2 (a) and (b) show some typical computational results of the quartic equations. The result in Figure 2 (a) has four distinct real roots, while that of Figure 2 (b) has two distinct real roots and two complex conjugates. The root which provides the shortest distance will be selected as the solution.

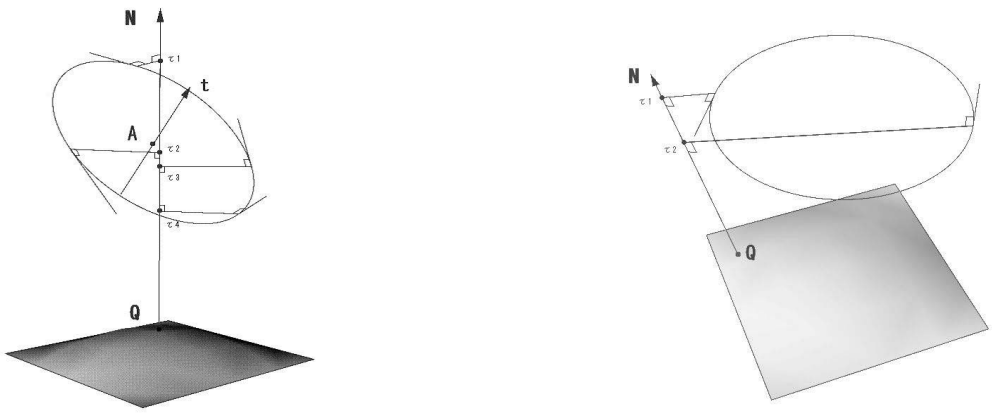
There are four cases when the determinant *Det* becomes zero, namely:

1. $\mathbf{N}$ is parallel to $\mathbf{t}$, i.e. $\mathbf{N} \parallel \mathbf{t}$ (see Figure 3 (a))

   Since $\mathbf{N} \parallel \mathbf{t}$, we have $\mathbf{n}\cdot\mathbf{N} = \mathbf{b}\cdot\mathbf{N} = 0$, and hence $Det = 0$. In this case the quartic equation reduces to a quadratic equation $(\tau-\varepsilon)^2 = 0$. Geometrically the distance computation between a line and a circle in 3*D* is reduced to a distance computation between a point and a circle in 2*D* as shown in Fig. 3 (a). Thus the distance vector is given by

$$\mathbf{d} = \mathbf{A} - (\mathbf{Q} + \tau\mathbf{N}) - \frac{\mathbf{A} - (\mathbf{Q} + \tau\mathbf{N})}{|\mathbf{A} - (\mathbf{Q} + \tau\mathbf{N})|}R = (\mathbf{A} - (\mathbf{Q} + \tau\mathbf{N}))\left(1 - \frac{R}{|\mathbf{A} - (\mathbf{Q} + \tau\mathbf{N})|}\right), \tag{21}$$

   where $\tau = \varepsilon = (\mathbf{A} - \mathbf{Q})\cdot\mathbf{N}$.



(a) (b)

Fig. 2. Roots of the quartic equation: (a) four distinct real roots, (b) two distinct real roots and complex conjugates

2. **N** intersects **t**, i.e. $\mathbf{Q} + \tau\mathbf{N} = \mathbf{A} + \xi\mathbf{t}$ (see Figure 3 (b))
   Since $\mathbf{t}\cdot\mathbf{n} = \mathbf{t}\cdot\mathbf{b} = 0$, we have $(\mathbf{Q} + \tau\mathbf{N} - \mathbf{A})\cdot\mathbf{n} = (\mathbf{Q} + \tau\mathbf{N} - \mathbf{A})\cdot\mathbf{b} = 0$ $(\alpha\tau - \gamma = \beta\tau - \delta = 0)$, and hence $Det = 0$. Under these conditions, the quartic equation can be written as
   $$(\tau - \tau_D)^2\left(\tau - \varepsilon - R\sqrt{\alpha^2 + \beta^2}\right)\left(\tau - \varepsilon + R\sqrt{\alpha^2 + \beta^2}\right) = 0\,, \qquad (22)$$
   where $\tau_D = \frac{\gamma}{\alpha} = \frac{\delta}{\beta}$. Therefore the roots are $\tau_1 = \varepsilon + R\sqrt{\alpha^2+\beta^2}$, $\tau_2 = \varepsilon - R\sqrt{\alpha^2+\beta^2}$, and a double root $\tau_3 = \tau_4 = \tau_D$ as shown in Figure 3(b). Note that $\tau = \tau_1, \tau = \tau_2$ are solutions to the governing equations (6), whereas the double root $\tau = \tau_D$, corresponding to the intersection point, makes the determinant become zero and does not satisfy (6).

3. **N** is perpendicular to **t**, i.e. $\mathbf{N}\cdot\mathbf{t} = \mathbf{0}$ (see Figure 3 (c))
   With this condition, it is easy to show that $\alpha^2 + \beta^2 = 1$, $\alpha\gamma + \beta\delta = \varepsilon$, and hence the determinant reduces to $Det = R(\tau - \varepsilon)$. In this case the quartic equation can be written as:
   $$(\tau - \varepsilon)^2\left((\alpha\tau - \gamma)^2 + (\beta\tau - \delta)^2 - R^2\right) = 0\,. \qquad (23)$$
   Apparently $\tau = \varepsilon$ is a double root, and with this root, the determinant becomes zero. However, unlike Case 2, the double root does satisfy (6).

4. **N** passes through **A**, i.e. $\mathbf{A} - \mathbf{Q} \parallel \mathbf{N}$ (see Figure 3 (d))
   Since $\mathbf{A} - \mathbf{Q} = \mathbf{N}\cdot(\mathbf{A} - \mathbf{Q})\mathbf{N}$, we have $\gamma = \varepsilon\alpha$ and $\delta = \varepsilon\beta$, and hence the determinant reduces to $Det = R(\alpha^2 + \beta^2)(\tau - \varepsilon)$. The quartic equation becomes the same as (22). Therefore this is a special case of Case 2 and similar discussions can be made.

3. CONTOURING METHOD

The tangent vector of the circle is given by differentiating Equation (1) with respect to θ, yielding

$$\dot{\mathbf{L}}(\theta) = R(-\sin\theta\mathbf{n} + \cos\theta\mathbf{b})\,. \qquad (24)$$

Since $\mathbf{d}\cdot\mathbf{N} = 0$ and $\mathbf{d}\cdot\dot{\mathbf{L}} = 0$ by definition (see Figure 1 (b)), and we have $(\mathbf{N}\times\dot{\mathbf{L}})\cdot\mathbf{N} = 0$ and $(\mathbf{N}\times\dot{\mathbf{L}})\cdot\dot{\mathbf{L}} = 0$ from the definition of the triple scalar product, we can conclude that the distance vector **d** is parallel

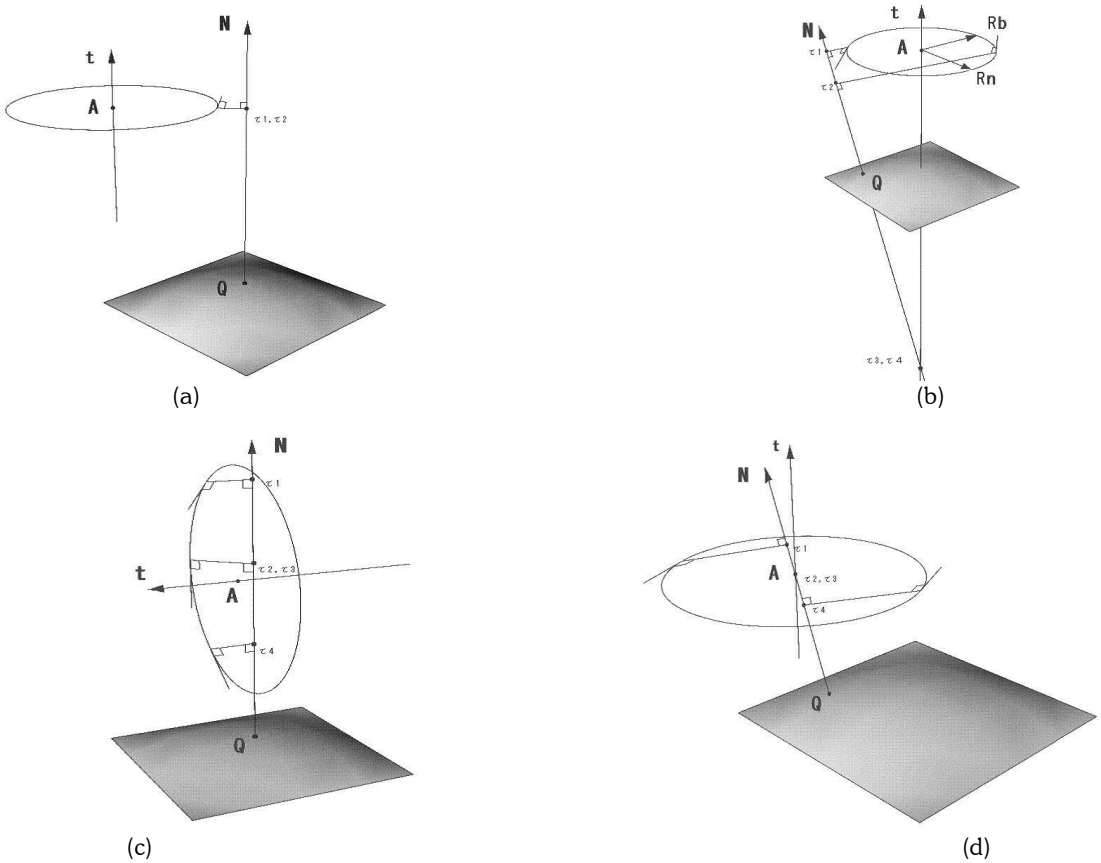


Fig. 3. Four cases when the determinant becomes zero: (a) **N** is parallel to **t**, (b) N intersects **t**, (c) **N** is perpendicular to **t**, (d) **N** passes through **A**

to $\mathbf{N} \times \dot{\mathbf{L}}$. A signed distance function d_s [1] can be defined by taking the dot product with the unit vector $\left(\mathbf{N} \times \dot{\mathbf{L}}\right) / \left|\mathbf{N} \times \dot{\mathbf{L}}\right|$ as follows:

$$d_s(u,v) = (\mathbf{A} + R(\cos\ \theta\mathbf{n} + \sin\ \theta\mathbf{b}) - (\mathbf{Q}(u.v) + \tau\mathbf{N}(u,v))) \cdot \frac{\mathbf{N}(u,v) \times \dot{\mathbf{L}}(\theta)}{\left|\mathbf{N}(u,v) \times \dot{\mathbf{L}}(\theta)\right|}. \quad (25)$$

Circular highlight lines are points on a surface where 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 [2] where curves of zero height are computed through surface-plane intersection problems as illustrated in Figure 4. Although the accuracy of 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 a given radius ρ, which is obtained by simply finding the points satisfying $d_s = \rho$ and $d_s = -\rho$ (see Figure 4). These two curves are called *circular highlight band boundary curves* and form a *circular highlight band* [1].

In the contouring algorithm [2], each rectangular grid is divided into four triangles by adding a new grid point at the intersection of two diagonals. The height at the new grid point is assigned a value corresponding to the average values of the four height values at the grid points. Therefore for each rectangular grid, four triangles are constructed in the *3D* space. These triangles are intersected by horizontal planes resulting in straight line segments, which form contour curves at the user specified contour heights.

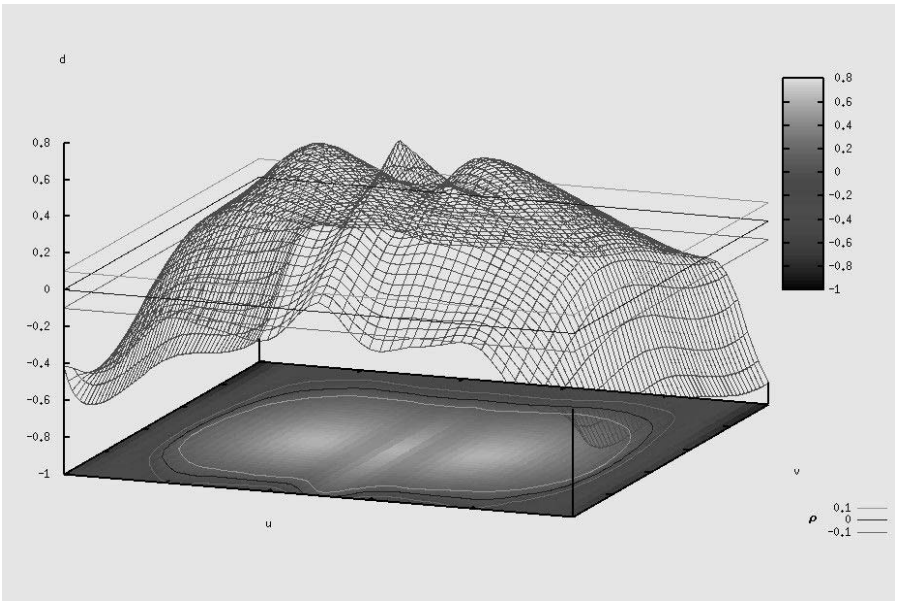


Fig. 4. A graph of signed distance function

4. COMPLEXITY ANALYSIS

We will develop an analysis of the complexity of the circular highlight line algorithm. The computation includes three parts, time on computing the signed distance at each grid point (T_1), the time on computing the contour lines (T_2), and the time on mapping the pre-image of the circular highlight lines to the 3D space (T_3). The time count on the first step (T_1) is obviously O(N), where N is the total number of grid points. Although the circular highlight line algorithm requires computation of solving the quartic equation, it has a closed form solution (see Appendix), and hence the additional computational cost is very small comparing to the highlight line algorithm [1]. It is easy to find that the time costs on the second step (T_2), as well as the third step (T_3), are also O(N). Therefore, the overall complexity of the circular highlight line algorithm is O(N) as in the highlight line case [1].

We have implemented the algorithm on an Intel Pentium 4 PC running at 3GHz. Figure 5 shows the computational time obtained by using the clock() function which returns an approximation of processor time used by the program. The surface used for the measurement is a bicubic B-spline surface shown in Figure 1. The grid resolution varies from 10×10 to 100×100. We can observe from Figure 5 that the computational time is linear and verifies our analysis.

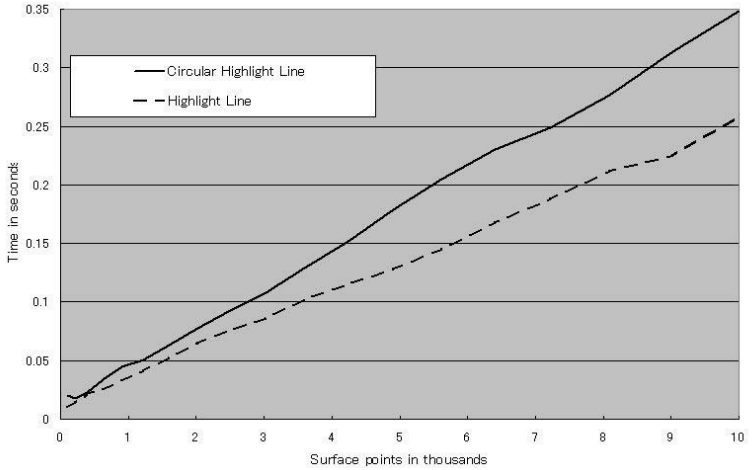


Fig. 5. Performance of the Circular Highlight Lines

5. CIRCULAR REFLECTION LINE MODEL

We can easily apply the techniques that we have developed for the circular highlight lines in Sections 2, and 3 for the computation of circular reflection lines. If we denote the eye position by **E** and the unit vectors $(\mathbf{E} - \mathbf{Q}(u, v))/|\mathbf{E} - \mathbf{Q}(u, v)|$ and $(\mathbf{L}(\theta) - \mathbf{Q}(u, v))/|\mathbf{L}(\theta) - \mathbf{Q}(u, v)|$ by **e** and **c** as shown in Figure 6 (a), we have the following three equations for **c** as follows:

$$\mathbf{c} \cdot \mathbf{N}(u,v) = \cos\alpha\,, \qquad \mathbf{c} \cdot \mathbf{e}(u,v) = \cos 2\alpha\,, \qquad |\mathbf{c}| = 1\,, \tag{26}$$

where $\cos\alpha = \mathbf{e} \cdot \mathbf{N}(u,v)$. Circular reflection lines can be computed by simply replacing the surface normal **N** in Sections 2 and 3 by the unit vector **c** obtained from (26). Figure 6 (b) shows circular reflection lines on a bi-cubic B-spline surface. A point with a bold letter **E** indicates the position of an eye. Comparing to the circular highlight line algorithm, the extra amount of computing cost is the calculation of vector **c**, which has a closed form solution. Therefore, the overall complexity of the circular reflection line algorithm is also *O(N)*. Note that this algorithm can be also applied to the conventional reflection line computation.

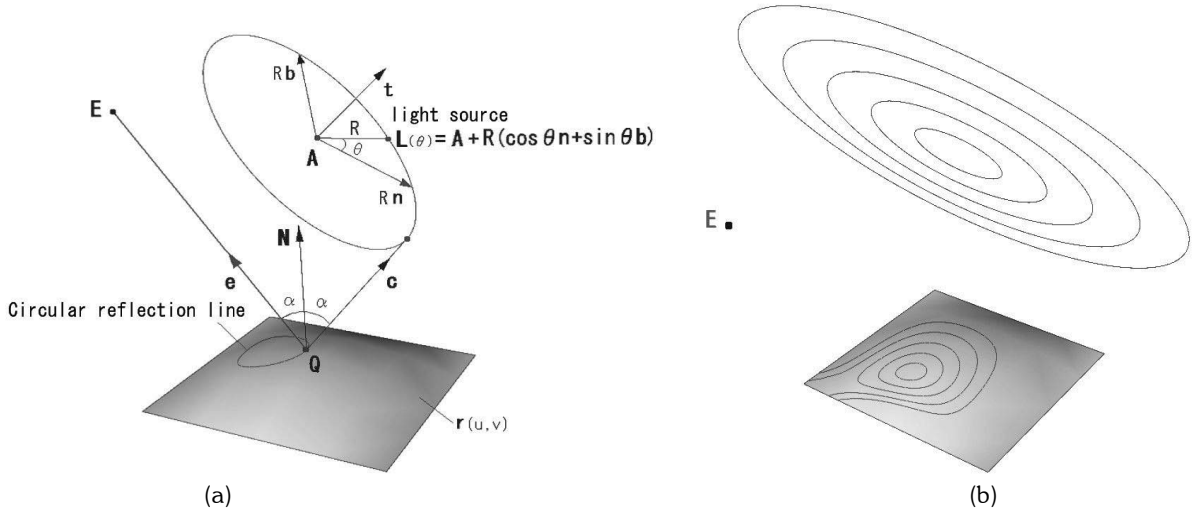
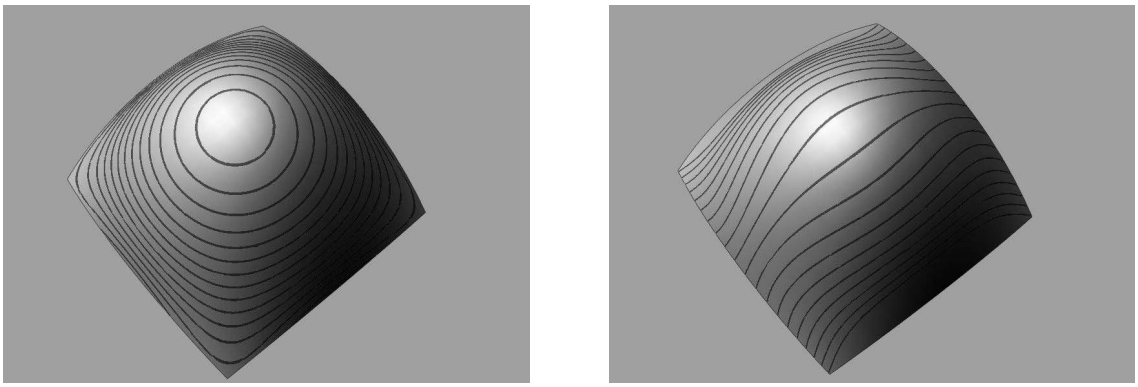


Fig. 6. (a) Definition of a circular reflection line. (b) Circular reflection lines on a bi-cubic B-spline surface.

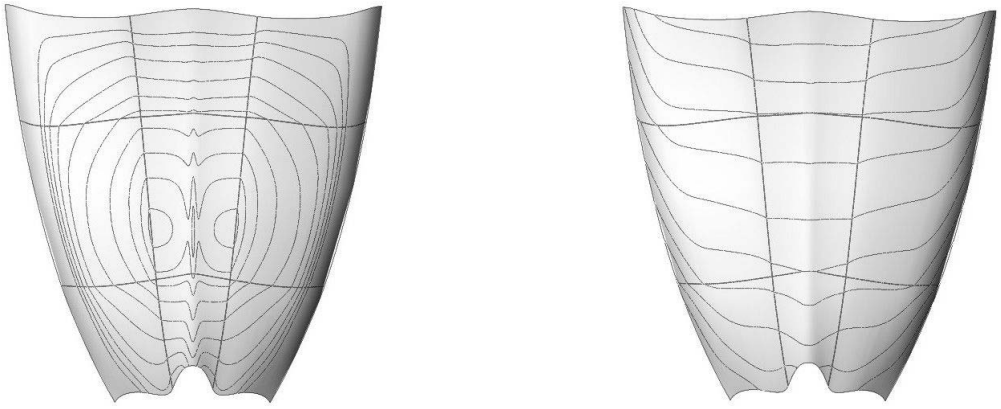
6. EXAMPLES

The circular highlight lines are most effective when the surface under interrogation consists of mainly elliptic points. We can observe from Fig. 7 (a) that the resulting circular highlight lines on a surface with mainly elliptic points show the fairness of the surface more clearly than that of the conventional highlight lines (b). Figure 8 (a) shows circular highlight lines on a hood of an automobile, while Figure 8 (b) shows conventional highlight lines on the same surface. The bi-cubic B-spline has a knot multiplicity of two at knots $u = 0.25$, $u = 0.75$, and $v = 0.25$, and $v = 0.75$, where $0 \le u, v \le 1$, therefore the surface is C^1 continuous at those knots. If the surface is C^1 continuous, then the circular highlight line will be C^0. It is clear from Figure 8 (a) that the resulting circular highlight lines can detect second order discontinuities in both u and v directions, while the conventional highlight lines are able to detect second order discontinuities only in u direction. Figure 9(a) and (b) depict circular reflection lines on a rear part of an automobile with different view points.



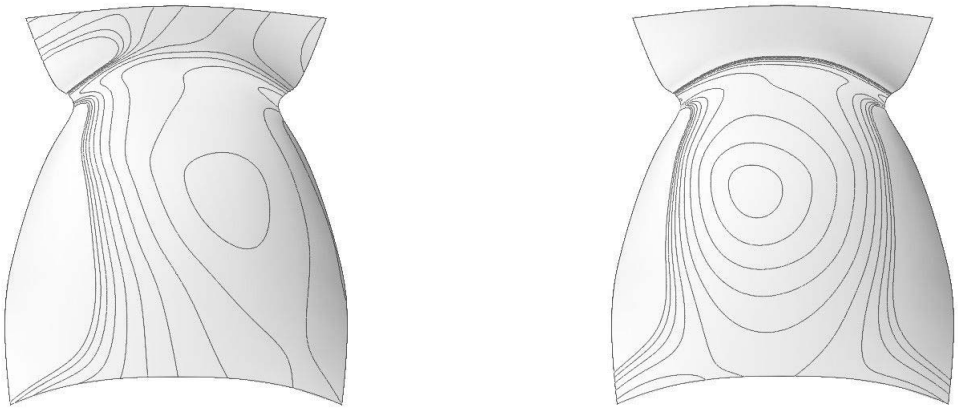
(a) (b)

Fig. 7. Circular highlight lines versus conventional highlight lines on a surface consisting of mainly elliptic points: (a) circular highlight lines, (b) conventional highlight lines



(a) (b)

Fig. 8. A hood surface of an automobile with C^1 continuity at knots $u = 0.25$, $u = 0.75$, $v = 0.25$, and $v = 0.75$, where these knots are depicted as iso-parametric lines in the figure: (a) circular highlight lines can detect second order discontinuities in both u and v directions, (b) conventional highlight lines can detect second order discontinuities only in u direction



(a) (b)

Fig. 9. Circular reflection lines on a rear part of an automobile with different view points.

7. CONCLUSION AND RECOMMENDATION

We have introduced a novel method for surface interrogation called circular highlight/reflection lines. Unlike the conventional first order surface interrogation methods such as, reflection lines and highlight lines, where a family of parallel straight lines are used for the light sources, concentric circle light sources are used for the newly proposed circular highlight/reflection lines to evaluate the surface fairness. In this way we can capture the surface fairness in all directions, opposed to the conventional methods that can capture the fairness only in one direction. The overall computational time cost of the circular highlight/reflection line algorithm is $O(N)$, where N is the number of grid points.

Acknowledgement

The authors would like to thank Nick Patrikalakis for letting us use the NURBS library developed at the MIT Design Laboratory.

Appendix: A solution of quartic equation based on Hacke's algorithm [5].

The general quartic equation (15) can be reduced to the standard form

$$x^4 + px^2 + qx + r = 0\,, \tag{27}$$

where

$$x = \tau + c_3/4c_4 \tag{28}$$

$$p = (-3c_3^2 + 8c_4c_2)/8c_4^2 \tag{29}$$

$$q = (c_3^2 - 4c_4c_3c_2 + 8c_4^2c_1)/8c_4^3 \tag{30}$$

$$r = (-3c_3^4 + 16c_4c_3^2c_2 - 64c_4^2c_3c_1 + 256c_4^3c_0)/256c_4^4 \tag{31}$$

Since we have

$$(x^2 + y)^2 = x^4 + 2x^2y + y^2\,, \tag{32}$$

for any value of y, we obtain

$$(x^2 + y)^2 = -px^2 - qx - r + 2x^2y + y^2 = (2y - p)x^2 - qx + (y^2 - r)\,, \tag{33}$$

by substituting for x^4 from (27). The right hand side of Equation (33) becomes a perfect square if the discriminant in x is zero, which leads to the following resolvent cubic:

$$q^2 - 4(2y - q)(y^2 - r) = 0\,, \tag{34}$$

$$8y^3 - 4py^2 - 8ry + 4pr - q^2 = 0\,. \tag{35}$$

Let y_1 be a real root of the resolvent cubic, then (33) becomes

$$(x^2 + y_1)^2 = K^2x^2 - 2KLx + L^2\,, \tag{36}$$

where

$$K^2 = 2y_1 - p\,, \qquad L^2 = y_1^2 - r\,, \qquad 2KL = q \tag{37}$$

Thus, we have the following two quadratics

$$x^2 - Kx + y_1 + L = 0\,, \qquad x^2 + Kx + y_1 - L = 0\,. \tag{38}$$

The roots of the quadratics are given by

$$x = \frac{K \pm \sqrt{K^2 - 4(y_1 + L)}}{2}\,, \qquad x = \frac{-K \pm \sqrt{K^2 - 4(y_1 - L)}}{2}\,. \tag{39}$$

8. REFERENCES

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

[2] Paul Bourke. CONREC A contouring subroutine. *http://astronomy.swin.edu.au/ pbourke/projection/conrec/*, July 1987.

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

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

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

[6] H. Hagen, S. Hahmann, T. Schreiber, Y. Nakajima, B. W¨ordenweber, and P. Hollemann-Grundstedt. Surface interrogation algorithms. *IEEE Computer Graphics and Applications*, 12(5):53–60, September 1992.

[7] T. Kanai. Surface interrogation by reflection lines of a moving body Bachelor's thesis, The University of Tokyo, Department of Precision Machinery Engineering, Tokyo, Japan, 1992. In Japanese. http://web.sfc.keio.ac.jp/ kanai/rline/bth.pdf

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

[9] T. Kraska. Subroutines for solving cubic, quartic and quintic equations. http://van-der-waals.pc.unikoeln.de/persons/kraskaE.html

[10] T. Maekawa, Y. Nishimura and T. Sasaki. Circular highlight / reflection lines. YNU TLO Disclosure Case 04041, Oct. 8, 2004. Patent pending No. 2005-003242, Jan. 7, 2005.

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

[12] J. Pegna and F. E. Wolter. Surface curve design by orthogonal projection of space curves onto free-form surfaces. Journal of Mechanical Design, ASME Transactions, 118(1):45-52, march 1996.

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

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

[15] H. Pottmann and J. Wallner. *Computational Line Geometry. Mathematics + Visualization* Heidelberg, Germany: Springer-Verlag, 2001.

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

[17] E. J. Wilczynski and H. E. Slaught. *College algebla with applications*. Allyn and Bacon, 1916

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