Efficient Algorithm for Real-time Generation of Reflection Lines

Juyup Kang
TM Development iBGEN Inc.
Kunwoo Lee, Sangkun Park, Tae-wan Kim*
Computer Aided Design and Analysis Lab., School of Mechanical and Aerospace Engineering, Seoul National University

Depending upon the method of the surface generation and the quality of the designed boundary curves, the resulting surfaces may have global or local irregularities in many cases. Thus, it would be necessary for the designer to evaluate the surface quality and to modify the surface. This is very important because the defect of the surface causes the rework of the dies, increasing cost and delivery time significantly.

To simulate the reflection line test in the actual production line, a faster algorithm for generating reflection lines is presented. In this paper, among various surface interrogation methods using reflection lines, Blinn-Newell type of reflection mapping is applied to generate the reflection lines on the trimmed NURBS surfaces. The derivation of reflection lines is formulated as a surface-plane intersection problem (Jung 1994) and is solved by surface-contouring techniques. Also, for eliminating the discontinuity of reflection lines due to the configuration of reflection map, a modified reflection map is proposed. An efficient traced contouring technique is utilized for the computational efficiency and proves to be well suited for the real-time quality-assessment task.

Key Words : Computer Aided Design, B-splines, Surfaces, Reflection Lines, Surface Interrogation

1. Introduction

Analyzing and interrogating designed surfaces remains an important issue being extensively studied in computer-aided geometric design. Surface design can be guided by various criteria, such as geometric conditions, manufacturing constraints, and aesthetic principles. The purpose is to create a smooth surface that captures the design intent and satisfies the constraint and performance criteria.

* Corresponding Author,
E-mail : tkim@cad.snu.ac.kr
TEL : +82-2-880-7447 ; **FAX** : +82-2-887-2283
Computer Aided Design and Analysis Lab., School of Mechanical & Aerospace Engineering, Seoul National University, Seoul, 151-742, Korea. (Manuscript **Received** April 10, 2000; **Revised** November 16, 2000)

The conventional process of surface fairing in automobile body styling can be described as follows. First a clay model is measured to yield the point clouds. From the point clouds many section curves are extracted, and the corresponding surface is generated by applying skinning operations to these section curves. Depending upon the method of the surface generation and the quality of the provided boundary curve, the resulting surfaces may have global or local irregularities. Thus, it is necessary for the designer to evaluate the surface quality and to modify the surface. This is very important because any surface quality defects in the production stage would require the rework of the dies, which is costly and time consuming.

Treating families of characteristic curves (Theisel and Farin, 1997) lines of curvature, asymptotic lines, isophotes, and reflection lines on the surfaces is a popular method of surface interrogation. Interrogation techniques attempt to illuminate curve and surface characteristics which cannot be easily discerned using a conventional rendering method. They usually exaggerate the shape to more easily detect a curve's or a surface's subtle flaws.

Reflection line method is used in the automobile industry to judge the reflection pattern of a series of parallel band lights. In the actual production line a whole car is located in a chamber having parallel band lights on its walls and ceiling, and the reflection of these band lights on the car surfaces are examined as a measure of its surface quality. If this reflection pattern is 'nice', then the aesthetic quality of the car body is considered satisfactory. Thus, it would be very helpful if these reflection patterns could be simulated in order to detect the surface irregularities before the surface is actually manufactured. To simulate the reflection line test in the actual production line as closely as possible, a faster generation algorithm is presented in this paper. Among various surface interrogation methods using reflection lines, Blinn-Newell type of reflection mapping (Blinn and Newell, 1976) is applied to generate reflection lines on the trimmed NURBS surface. The reflection lines calculated from the reflection mapping requires a simple calculation but conforms to the physical principle well enough.

The generation of reflection lines is formulated as a surface-plane intersection problem, which can be solved by surface-contouring techniques. An efficient traced contouring technique is utilized for the computational efficiency and proves to be well suited for the real-time quality-assessment task. Thus, the surface quality evaluation using reflection lines on graphics display can replace the real reflection test on the real car body in the light chamber. This will be an important element in realizing a virtual product design system for the automobile industry.

We begin our discussion by reviewing a reflection line algorithm based on reflection mapping. The contouring method is introduced as the workhorse of the proposed algorithm. The model of highlight lines is introduced to demonstrate the contouring algorithm in a simple way. A modified reflection map is then proposed for reliable results in surface interrogation. The preprocessor and line generation processes are also described. Finally, some examples and the performance analysis on implementation results are provided. The flow chart of the entire procedure for reflection line generation is shown in Fig. 1.

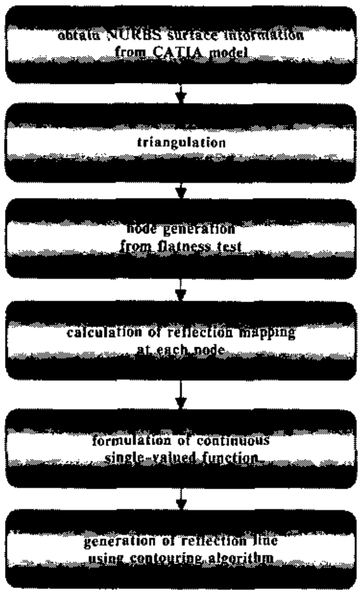


Fig. 1 Flow chart of proposed algorithm

2. Related Works

The reflection line method is used in the automobile industry to judge the reflection pattern of a series of parallel band lights. If this reflection pattern is 'even', then the aesthetic quality of the car body is considered satisfactory. It would be very helpful if these reflection patterns could be simulated in the design phase in order to detect surface irregularities before the surface is actually manufactured. There have been many approaches to generate reflection lines as described below.

Klass (1980) defined reflection lines in the following statement. A family of straight lines can be reflected onto a surface as seen from a fixed point of vision. Curves generated in this way on the surfaces are called reflection line. He proposed to calculate the reflections of the straight lines onto a surface by using numerical methods. This approach has a drawback in computational efficiency, because solving equations to get reflection points for multiple parallel line lights requires extensive calculations. Even for a single reflection point, estimating an initial value for the numerical solution is not simple.

A modified reflection line is proposed by Kaufmann and Klass (1988) for computational efficiency. In their method, a family of planes that are parallel to a common vector is intersected with a given C^2-continuous surface to be evaluated. Then C^2-continuous splines are obtained as the intersection curves. The corresponding points on the separate splines where the tangent vectors have the same angle with respect to a given vector are connected to result each reflection line. This modified definition of the reflection line requires less computation, but does not simulate the reflection line realistically. Above all, this modified reflection line is not affected by the location of the viewer.

Farin (1985) defined the reflection line as the contour of the directional derivative of the surface function with respect to a given direction; he solved it using numerical methods, but his method does not simulate the real reflection line either.

Reflection mapping was developed to model specular inter-object reflection by Blinn and Newell (1976). A center of projection, - usually the approximate centroid of the objects to be rendered-is chosen and used as the center of a virtual sphere surrounding the objects. Then the environment is projected onto the surface of the virtual sphere. This virtual sphere can then be treated as a 2D texture map, i. e. reflection map. At each point on an object to be displayed, the corresponding location in the reflection map is found by reflecting the view vector about the normal vector of the object at the point.

For the purpose of applying reflection mapping to the generation of reflection lines, Choi and Lee (1996) replaced the spherical environmental map with a virtual box having striped line patterns on each face in order to simulate parallel band lights in the light chamber. This makes it possible to simulate reflection tests for both parallel band and line lights. Their approach will be explained in detail, because the approach proposed in this paper is an improvement upon theirs. Their algorithm to generate reflection lines from reflection mapping consists of the following five steps.

Step 1

A reflection map consisting of colored stripes on a rectangular box is constructed as shown in Fig. 2. The box should be large enough to completely surround the surface under evaluation. Striped lines on the map play a role as the parallel band lights and their boundaries do as line lights. The difference between the parallel band light and line light is that the former has the width while the latter does not. The resulting reflection map box contains information about equations of planes for the walls, top, bottom, number of striped lines, and colors of each striped band for all six faces. The width of the stripe is such that the colored and uncolored stripes have the same width. The reflection map box made this way has the following advantages. It is easy to compare the straight-line image on the map with the distorted mapped image on the subject surface. In addition, since the intersection calculation between line and plane is required only, the computation involved in the mapping is simple.

Step 2

The surface is divided into triangular facets, since the mapping process is carried out for each

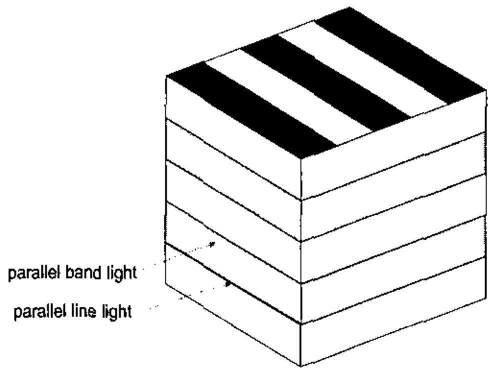


Fig. 2 Reflection map box (Choi and Lee, 1996)

triangular facet. The size of the facet does not have to be very small, because large facets will be refined later in the process of generating reflection lines. Since this triangulation is performed in the (u, v) parametric domain of the surface, each facet has three two dimensional (u, v) points for their vertices.

Step 3

For each facet, the reflection mapping process begins with the determination of colors for its three vertices. The color of a vertex, i. e. the color of the corresponding point on the reflection map, is determined such that the line between the corresponding point and the vertex of interest and the line between the vertex and the viewpoint should both lie in the same plane. They should make the same angle with respect to the surface normal vector at the vertex as shown in Fig. 3. This is a reasonable assumption conforming to the physical principle. Reflection lines obtained by this method vary as the designer moves around in the light chamber.

Step 4

The color of each facet is determined from the colors of its vertices. Based on the uniformity of the vertex colors, all the situations can be categorized into three cases as shown in Fig. 4.

Case 1: Three vertices of a facet have the same color. In this case, the corresponding color is assigned to the entire facet. The smaller the facet size is used, the more accurate result is produced.

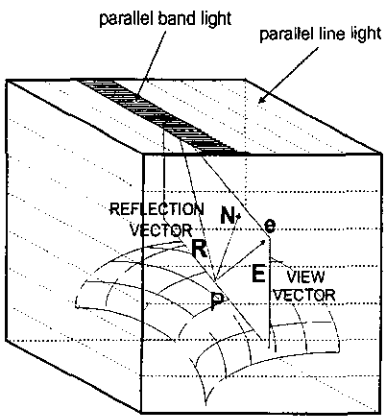


Fig. 3 Determination of vertex color

We can find a small facet where the corresponding region in the reflection map can be a triangle and the triangle resides in the same stripe when all of its vertices are inside the stripe.

Case 2: If the color of one vertex is different from the other two vertices, the facet should be divided into four facets f_1, f_2, f_3 and f_4 as shown in Fig. 5. Deleting the facet and inserting f_1, f_2, f_3 and f_4 instead can carry out this process. In Fig. 5, point b_3 is simply the mid-point of v_1 and v_2 in the parametric domain. In this example, v_1 and v_2 are assumed to have the same color. Point b_1' can be obtained from the intersection between the line $v_1'v_3'$ and the boundary line L on the map. Here v_1' and v_3' are the mapped points of points v_1 and v_3 onto the reflection map, respectively. Once the intersection point is obtained on the reflection map, the u, v parameters of the corresponding points b_1 and b_2 of the facet are derived from Eq. (1) and Eq. (2).

$$\binom{u}{v}_{b_1} = \frac{|v_1' - b_1'|\binom{u}{v}_{v_3} + |b_1' - v_3'|\binom{u}{v}_{v_1}}{|v_1' - v_3'|} \quad (1)$$

$$\binom{u}{v}_{b2} = \frac{|v_2' - b_2'|\binom{u}{v}_{v_3} + |b_2' - v_3'|\binom{u}{v}_{v2}}{|v_4' - v_2'|} \quad (2)$$

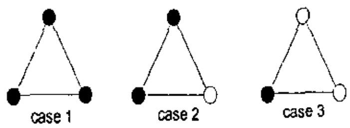
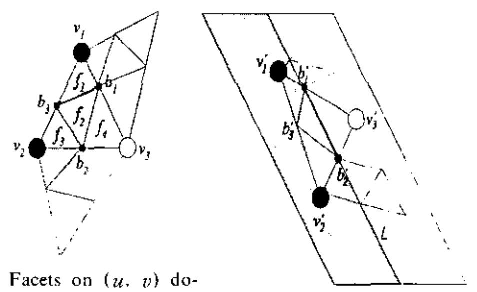


Fig. 4 Categorization based upon uniformity of vertex colors (Choi and Lee, 1996)



(a) Facets on (u, v) domain of the surface (b) Facets on the reflection map

Fig. 5 Subdivision of a facet into four facets (Choi and Lee, 1996)

In the equations above, the term with the bracket is the length of the line segment between two points inside the bracket.

After the subdivision, the facets f_1, f_2 and f_3 are assigned the color of the vertex v_1 while facet f_4 is assigned the color of vertex v_3.

Case 3: If all the vertices have different colors, the facet should be subdivided until all the subdivided facets can be categorized under case 1 or case 2.

Step 5

Once reflection mapping is performed for all the facets, the result for either case of parallel band lights and line lights can be displayed. For the parallel band lights, the facet is colored with the color obtained in step 4. For the line lights, their reflected image can be easily displayed by drawing all the line segments, which is on the boundaries of parallel bands.

3. Contouring Algorithm

Choi and Lee generated reflection lines by calculating the mapping of boundaries of parallel band lights onto all triangular facets. This mapping process together with the subdivision process requires heavy computation. Instead of these time-consuming steps, contouring algorithm is applied to find reflection lines directly. Time consumption of the contouring process is relatively small and enables to update the reflection lines in real time as the location of the viewer changes. It is also robust and fully automatic.

The contouring algorithms are used to find curves of constant height (isolines or contour lines) on a single-valued surface. The surface is usually defined by discrete values given at the vertices of a mesh. The mesh can be defined arbitrarily in the parametric uv space of the surface. The accuracy of the contouring process can be controlled through the mesh density. The concept of the contouring algorithm will be explained with the previous work in which highlight lines of a surface is calculated.

Beier and Chen (1994) defined the highlight line as a set of points on a surface for which the perpendicular distance between the surface normal vector and a linear light source is zero. Thus, the condition for a surface point belonging to a highlight line can be formulated as follows. Assume the linear light source $L(t)$ (see Fig. 6) is given by

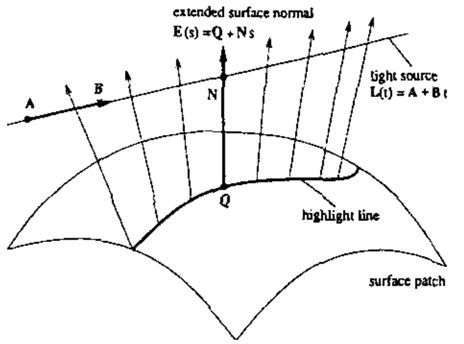


Fig. 6 Definition of highlight line (Beier and Chen, 1994)

$$L(t)=A+B\cdot t \tag{3}$$

where **A** is the position vector of a point on $L(t)$ and **B** is a vector defining the direction of $L(t)$. For a given surface point Q, let N be the directional vector of the corresponding surface normal. The extended surface normal $E(s)$ at Q is, therefore, a line passing through Q in the direction of N, and it can be defined as follows:

$$E(s)=Q+N\cdot s \tag{4}$$

Point Q belongs to the highlight line if both lines $L(t)$ and $E(s)$ intersect, that is, if the perpendicular distance d between both lines given by

$$d=\frac{|(B\times N)\cdot(A-Q)|}{\|(B\times N)\|} \tag{5}$$

is zero.

The calculation of highlight lines on a parametric surfaces $Q(u, v)$ can be formulated as the surface-plane intersection problem performed on an offset surface defined by the distance d over the parametric domain of the original surface. The perpendicular distance d between a surface-normal direction and linear light-source can be calculated for any surface point $Q(u, v)$. Therefore, it is treated as a continuous function $d(u, v)$. This distance function $d(u, v)$ is a single-valued surface over the uv domain as shown in

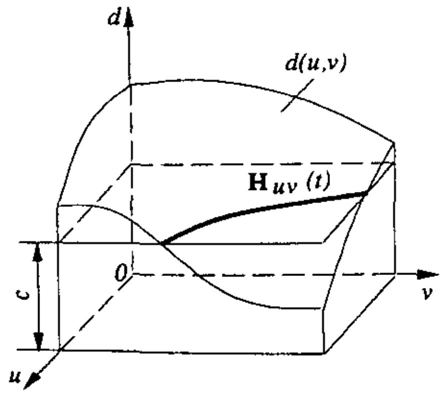


Fig. 7 Offset surface (Beier and Chen, 1994)

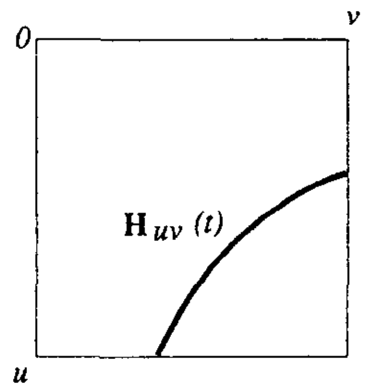


Fig. 8 Contour line in u, v domain (Beier and Chen, 1994)

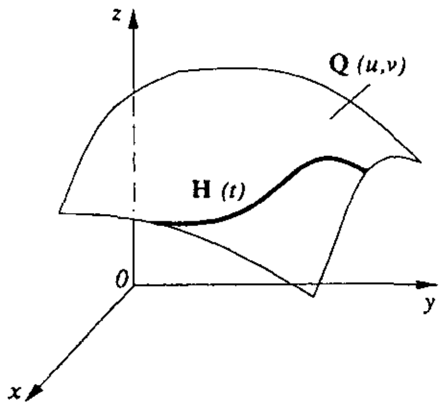


Fig. 9 Contour line in real domain (Beier and Chen, 1994)

Fig. 7. When intersecting this offset surface with a plane $d=c$ parallel to the uv plane, one or more intersection curves $\boldsymbol{H}_{uv}$ are obtained; this can be projected in the uv plane as shown in Fig. 8. Using a surface evaluator for $\boldsymbol{Q}(u, v)$, the intersection curves can be mapped into the object space of the surface $\boldsymbol{Q}(u, v)$, as shown in Fig. 9, and a highlight line is obtained for $c=0$.

Previous explanations imply that the highlight-line model is a simplified reflection model. The elimination of the viewpoint decouples the viewing operation from the manipulation of highlight lines. Therefore, it is impossible to simulate the movement of the observer in the actual reflection line test.

4. Modification of Reflection Map

The reflection lines, as visual smoothness indicators, are meaningful in that the discontinuous reflection lines can identify local surface imperfections. However, discontinuities of reflection lines due to the configuration of reflection map can mix with the wiggled reflection lines caused by local surface irregularities. As a result, the designer may not be able to identify the locations of surface imperfections. Thus, a modified reflection map is necessary to prevent discontinuities due to its configuration.

To apply contouring methods to generate reflection lines, the offset values should be assigned properly to each node in uv parametric domain and the offset surface may be constructed accordingly. These offset values should be assigned to represent the associated color stripe. Since reflection lines correspond to boundaries of the color stripes of the map box, each boundary of the color stripe should have a distinct offset value. Therefore, any reflection line is obtained by intersecting the offset surface with a plane with a proper constant offset value. One natural way to determine the offset value at any node in uv domain would be to utilize the coordinate of the intersection between the reflection vector and the plane of reflection map box. Recall that the reflection vector and corresponding intersection point are illustrated in Fig. 3. If four sidewalls are considered only the z coordinate provides information on the band lights to which the intersection point belongs. The pattern of the stripes and coordinate system is shown in Fig. *10*. For the top

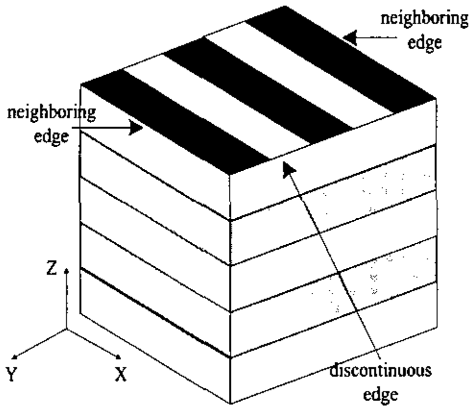


Fig. 10 Edges where discontinuities occur in reflection map

face and bottom face the y coordinate does the equivalent role. Thus, the offset surface may be constructed from these coordinate values if the offset values are continuous over *uv* domain in the surface of interest. However, the offset values determined this way can not be continuous across the discontinuous edge between the upper and side faces (see Fig. 10). Even across the neighboring edges shown in Fig. 10, the offset values can not be continuous across both edges. This can be explained as follows. As mentioned earlier, the z coordinate of the intersection point is taken as the offset value if the intersection point is on a sidewall. Thus, the y coordinate of the intersection point on the top face (or bottom face) should be shifted such that its value near from one of the neighboring edges is close to the z coordinate at the side walls (height of the box is added for the top face). This results in a continuous single-valued function. However, within the top face (or bottom face), the offset value or the y value increases or decreases along the direction toward the opposite neighboring edge. The y value at the opposite neighboring edge can not be the same as the z value of the sidewall.

To overcome this discontinuity problem, a modified reflection map is proposed as shown in Fig. 11. Note the modification of the color stripes on the top face and the bottom faces. The proposed map has no discontinuous edges, and continuous offset values across the neighboring edges can be provided.

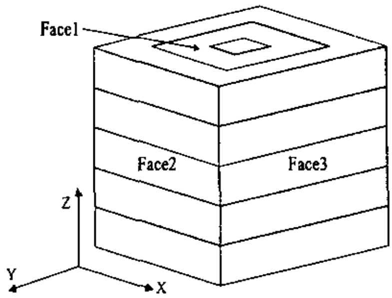


Fig. 11 Modified reflection map

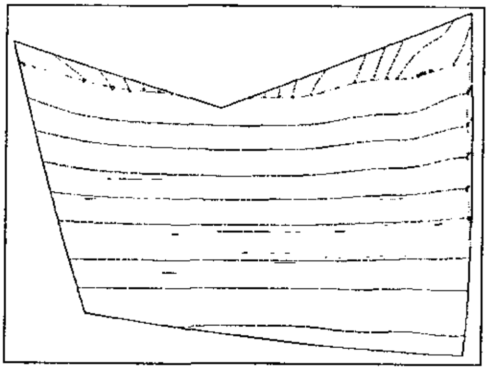


Fig. 12 shows the reflection lines generated from the original reflection map.

Figure 12 shows the reflection lines generated from the original reflection map. One can find discontinuous reflection lines across the discontinuous edge. This discontinuity does not result from surface irregularities but from the configuration of the reflection map. Figure 13 shows the reflection lines generated by the modified reflection map. There are no discontinuous lines resulting from the configuration of the map box. The darker line indicates the reflection line generated from the neighboring edge.

5. Implementation

5.1 Preprocessing

The reflection mapping calculations are performed at each node before running the contouring algorithm. Nodes for the reflection mapping are derived by triangulating the trimmed surface of interest in its parametric domain. During triangulation meshes are subdivided until they pass a flatness test. The preprocessing process is as fol-

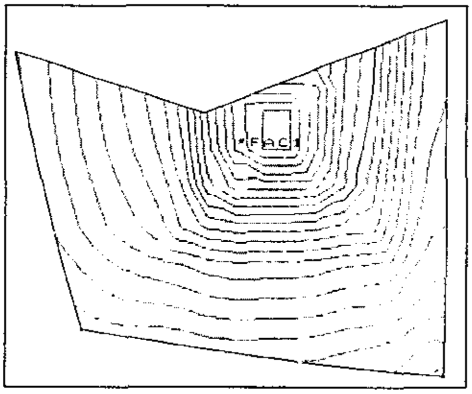


Fig. 13 Reflection lines generated by modified reflection map

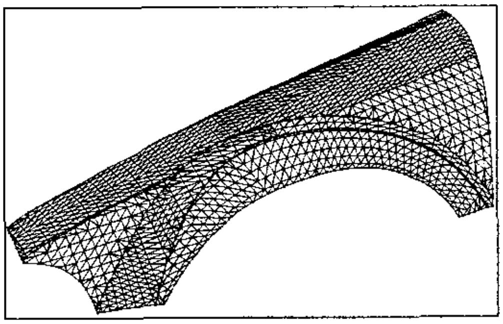


Fig. 14 Triangular mesh without flatness test

lows.

All the processes are developed under a CATIA API environment. CATIA was chosen because it is one of the most popular CAD systems in the automobile industry. Firstly, the information on the NURBS equation is obtained from the CATIA model file. After obtaining the NURBS information from the CATIA model file, a triangular mesh structure is prepared for the generation of the reflection lines. The CATGEO (Geometry Interface) function provides triangular mesh structures. CATGEO is a set of subprograms used to add, modify, or read basic data in CATIA. The position of nodes generated by CATGEO is based on iso-parametric lines, and these triangular facets do not reflect the geometric shape of the surface. Contours generated from rough mesh points will not be smooth. Thus, a simple flatness test and subdivision are applied to the facets. This test compares the minimum distance between the middle point of a triangular facet and the original surface with a user-specified tolerance. If the region around this facet is nearly flat, this distance would be small. If the distance is greater than the tolerance, the facet is subdivided by inserting a new node at the midpoint of each edge of the facet. Figure 14 shows the triangular meshes generated without the flatness test while Fig. 15 shows the meshes with the test. Notice that the meshes are dense in the region where the surface normal vector varies greatly.

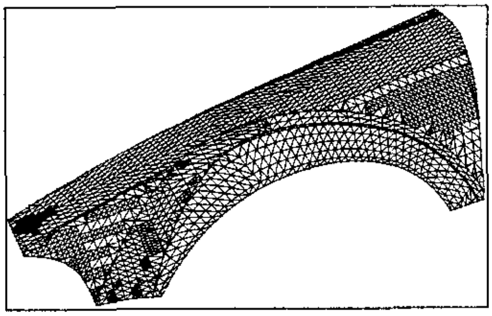


Fig. 15 Triangular mesh with flatness test

Once the triangular meshes are generated, they are stored in the edge-oriented data structure to be used in the contouring algorithm. This data structure includes the coordinates of the intersection point from the reflection mapping and the information of nodes, edges and facets. Information on the coordinates of the intersection points is not prepared during preprocessing since its values change whenever the viewing operation displays the reflection lines.

5.2 Generation of reflection lines

For each mesh point the reflection mapping process is performed to find the intersection point between the reflection vector and the face of the map box as shown in Fig. 3. The coordinate of the corresponding point on the reflection map is determined by the following steps.

1^{st} step: Find the position vector **P** of the given vertex as shown in Fig. 16.

2^{nd} step: Calculate the view vector **E** from the vector equation $\mathbf{E}=\mathbf{e}-\mathbf{P}$, where **e** is the position vector of the viewpoint.

3rd step: Calculate the normal vector **N** of the given surface at **P**

4th step: Find the relative reflection vector **R** which satisfies the following two conditions. Relative reflection vector **R** must be on the same plane composed of **N** and **E**. The angle between **N** and **E** must be equal to the angle between **N** and **R**.

5th step: Find the relative reflection line **L** that passes through **P** and whose directional vector is **R**.

6th step: Calculate the intersection point between **L** and the reflection map box. This point can be obtained from the plane equations of the map and the line equation of **L**.

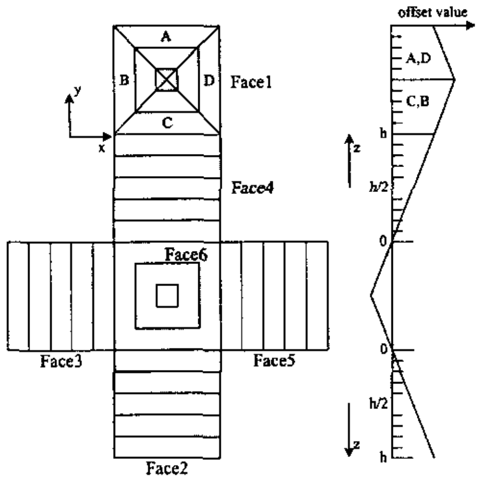


Fig. 16 Expanded view of reflection map box

From the calculated intersection points the offset surface is generated for the contouring process. This surface must not have different offset values for each band light. The offset values across the neighboring edges must be continuous. The modified reflection map shown in Fig. 16 satisfies these two conditions. For a given surface mesh and the results of the reflection mapping, the offset surface is constructed as follows.

Figure 16 shows how to obtain the offset values from the intersection point located at any face of the reflection map. For the four sidewalls (Face 2, Face 3, Face 4, and Face 5), the z coordinates of the intersection point are taken as the offset value. However, if the intersection point is on the top (Face 1) and bottom (Face 6) face, its offset value is determined by adding/subtracting the distance between the intersection point and its boundary edge to/from the offset value at the boundary edge. If the maximum z coordinate value of the wall faces is h, the offset value at the boundary edges between the top face and the wall faces is set to h, too. Then the offset value in the top face is obtained by adding h to the distance between the intersection point and the boundary edges of the upper face. In this way, continuous offset values are assigned across the boundary edges. The distance in the y direction is added to h when the intersection point is in region A and C (see Fig. 16), and the distance in the x direction is added in region B and D. It can be readily seen that the values in region A, B, C, and D are contiguous with the values in Face 2, Face 3, Face 4, and Face 5, respectively. For the bottom face, the offset value is obtained by subtracting the distance between intersection point and its boundary edge from the minimum z coordinate value of the wall faces. The graph of the offset value is at the right side of the expanded view of the reflection map.

Numerical contouring algorithms are frequently used in computer graphics applications. For the generation of reflection lines, the contouring algorithm is the workhorse', and is instrumental in realizing real time response. A speed-optimized contouring algorithm, NISO, developed at the University of Michigan was adopted (Piegl and Tiller, 1998).

NISO generates the contour lines of a single-valued continuous surface in linear time using the traced contour-line scheme. The surface is represented by its offset values at the vertices of a mesh which consists of quadrilateral faces and/or triangular faces. The set of interrelated tables for the triangular meshes is provided in the proposed approach. The vertex table lists all the vertices of the mesh with the corresponding uv coordinates and surface offset values. The edge table contains all the edges of the mesh. The face table defines all the faces of the mesh through pointers to their bounding edges-there are three edges for a triangular mesh. The edge and face tables contain only topological information about the mesh. Both tables can be created before the surface offset values are evaluated and registered in the vertex table. When an observer moves to a new position, the same edge and face tables can be used as long as the same mesh is maintained. This is one of the features that enables real time simulation.

5.3 Evaluation of reflection line generation algorithm

The algorithm was implemented on an Indigo2 workstation from Silicon Graphics, Inc. (R10000, Maximum impact). The efficiency of the algorithm comes from its linear-time-complex nature for surface-quality assessment. Although the construction of topology information tables and the preprocessing are time consuming, the generation of reflection lines according to the movement of viewer position and the reconfiguration of reflection map only takes about two seconds for trimmed surfaces with approximately two thousands nodes. The total time consumption of the algorithm can be divided into two major parts: the time for performing numerical computations and the time for displaying the results on the monitor. The computational part of the algorithm includes the calculation of the reflection mapping and the contouring process. The time consumed when displaying the graphics generally varies with the hardware. Note that the creation of CATIA geometric elements, i.e., node points, line segments, takes some additional time. Thus, the speed of display is not fast compared to the computational speed. However, the overall speed of the algorithm is good enough for the simulation of real reflection tests in the light chamber.

To verify the speed of the algorithm and the time complexity, the wall clock time was measured while varying the viewer position and the configuration of the reflection map. The experiment surfaces are the trimmed surfaces of a fender model as shown in Fig. 18. Figure 17 shows the response time for manipulation of the reflection lines. The horizontal axis indicates the number of nodes, which can be controlled in the preprocessing step. The data points labeled 'Mapping' represent two tasks: 1) reading files containing the information on the generated nodes and the triangular meshes and 2) calculating reflection mapping. Thus, the time complexity of this task is linear. The data points labeled 'Contouring' shows the time for the contouring process. The choice of the contouring algorithm in the generation of reflection lines proves to be excellent in that the speed of the contouring process is faster than any other processes. The data points labeled 'Line' indicate the time for displaying reflection lines on the monitor. From the set of points as a

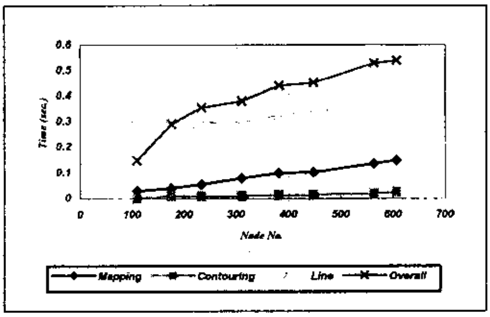
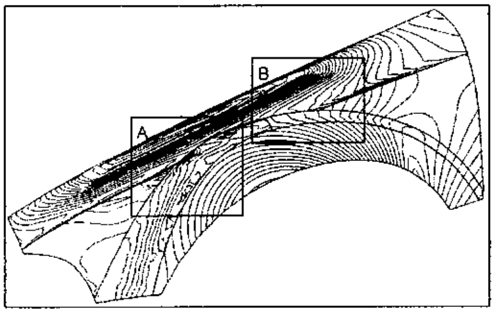
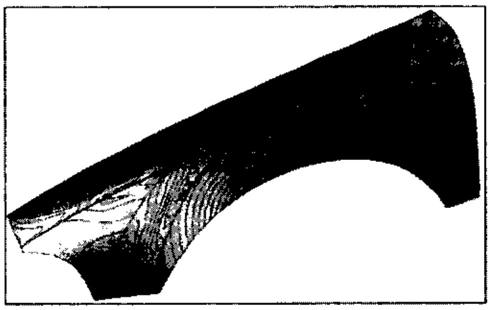


Fig. 17 Wall clock time of each operation



(a) Wire-frame image



(b) Shaded image

Fig. 18 Reflection lines on surfaces constituting fender

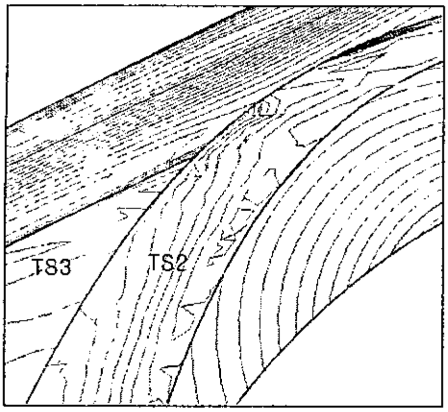


Fig. 19 Magnified view of (a) in Fig. 18

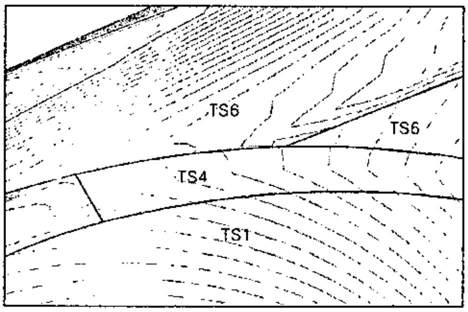


Fig. 20 Magnified view of (b) in Fig. 18

result of the contouring process, line segments are registered as CATIA line elements and displayed by CATIA.

6. Case Study

Figure 18 shows the simulated reflection lines on the surfaces constituting a car fender part. This part has six trimmed NURBS surfaces. The number of band lights is forty in this case. From this result the surface irregularities in areas A and B can be identified. The regions around A and B are magnified in Fig. 19 and Fig. 20. Figure 19 shows the ill-conditioned reflection lines in the trimmed surfaces TS2 and TS3, indicating that there is a drastic variation in the surface normal vector in TS2 and TS3. The discontinuous reflection lines across trimmed surfaces in Fig. 20 indicate the discontinuous connections between the trimmed surfaces. This discontinuity can especially be noticed at the boundaries between TS4 and TS6 (or TS5). Figure 21 shows the reflection lines on

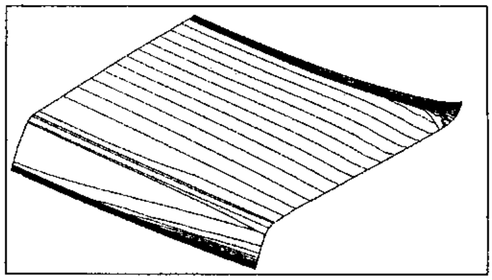


Fig. 21 Reflection lines on surfaces constituting trunk

the surfaces constituting a trunk for the line lights. This model consists of fifteen trimmed NURBS surfaces. From this result a designer can conclude that there are no surface irregularities in both surface quality and surface continuities.

From the result of the reflection lines, the global behavior of the curvature and the normal vector of the test surface can also be interpreted. From Fig. 18 it can be induced that the magnitude of the curvature is relatively small in regions where the gaps between reflection lines are large. The wiggled reflection lines indicate that the normal vector of the subject surface varies irregularly.

7. Conclusion

A speed-optimized reflection line algorithm for the simulation of a real reflection test is proposed. The proposed approach requires only a simple calculation compared to existing methods, but conforms to physical principles by calculating the reflection lines from the reflection mapping. In addition, a modified reflection map is proposed to prevent the discontinuity of reflection lines due to the configuration of the reflection map.

By simplifying the computation of reflection lines into a surface-plane intersection problem, the reflection lines can be easily updated with movements of the viewer position, reconfigurations of the reflection map, or different light sources. Thus, evaluation of surface quality using the reflection lines by the proposed approach could replace the real reflection test which requires the real car body surface.

Acknowledgements

The authors acknowledge the financial support from National Research Laboratory grant from The Korean Ministry of Science and Technology.

References

Beier, K.-P. and Chen, Y., 1994, "Highlight-line Algorithm for Real-time Surface-quality Assessment," *Computer Aided Design*, Vol. 26, No. 4, pp. 268~277.

Blinn, J. F., and Newell, M. E., 1976, "Texture and Reflection in Computer Generated Images," *Communications of ACM*, Vol. 19, No. 10, pp. 542~547.

CATIA Application Programming Interface Reference Manual.

Choi, I. and Lee, K., 1996, "Evaluation of Surfaces for Automobile Body Styling," *Computer Graphics International '96*.

Farin, G., 1985, "A Modified Clough-Tocher Interpolant," *Computer Aided Geometric Design*, Vol. 2, No. 1-3, pp. 19~27.

http://www.catia.ibm.com/prodinfo/prdfctn.html, CATIA Technical Support and CATIA Products by Function.

http://www.icem.com/products/surf/surf.htm, ICEM Surf.

Jung, H. B., 1995, "A Method for Surface/Surface Intersection," KSME Journal, Vol. 9, No. 1, pp. 51~67.

Kaufmann, E. and Klass, R., 1988, "Smoothing Surfaces Using Reflection Lines for Families of Splines," *Computer Aided Design*, Vol. 20, No. 6, pp. 312~316.

Klass, R., 1980, "Correction of Local Surface Irregularities Using Reflection Lines," *Computer Aided Design*, Vol. 12, pp. 73~77.

Lee, K. , 1999, "*Principles of CAD/CAM/CAE Systems*," Addison Wesley.

Piegl, L. A. and Tiller, W., 1998, "Geometry-based Triangulation of Trimmed NURBS Surfaces," *Computer Aided Design*, Vol. 30, No. 1, pp. 11~18.

Preusser, A., 1984, "TRICP: A Contour Plot Program for Triangular Meshes," *ACM Trans. Math. Soft.*, Vol. 10, No. 4, pp. 473~475.

Sutcliffe, D. C., 1980, "*Contouring over Rectangular and Skewed Rectangular grid An Introduction*," Mathematical Methods in Computer Graphics and Design, Academic Press, pp. 39~62.

Theisel, H. and Farin, G., 1997, "The Curvature of Characteristic Curves on Surfaces," *IEEE Computer Graphics and Application*, Vol. 17, No. 6, pp. 88~96.