# An Algorithm for Interpolation of Free-Form Surfaces

Akio Yajima*, Hirohumi Jonishi* and Junji Tsuda**

There are many successful methods of representing free-form surfaces. Most of these methods require setting many control points or boundary conditions in order to generate the surface which the designer wishes. A significant advantage of our proposed algorithm is the designer-oriented surface definition that is achieved, also it is only necessary to refer to the shapes of the boundary curves, usually represented in sketches or drawings, in order to construct the shape of the objects. Therefore, merely by using the information provided in engineering drawings, the designer can make composite free-form surfaces.

Our method is basically an extension of the Coons algorithm. Rotation and scaling operations are added to the Coons algorithm in order to obtain the surface shapes specified by the designer. The algorithm makes it possible to generate sculptured surfaces and highly-curved surfaces like tori, spheres, cones and so forth under several conditions which allow the designer to control the interior of the surfaces. The algorithm has been implemented for a CAD/CAM system for plastic molding dies. This system can be successfully applied in making practical products, e.g., consumer electrical appliances.

## 1. Introduction

Surface representation techniques have been investigated by a number of researchers mainly for developing CAD/CAM systems. Among these are Coons [2], Ferguson [8], Bezier [1], Riesenfeld [5] etc. For these techniques, it is assumed that a mesh of control points or curves has to be digitized from clay models of the objects which are made by skilled hands, and that interactive computer graphics systems should be used to form smooth shapes. These processes, however, involve a considerable amount of time and require considerable skill on the part of the designer, as the final shape must be formed by controlling or moving points in three-dimensional space. As a result, it is difficult for the designer to utilize these systems easily and effectively. In conventional designs, design and manufacturing information are provided on engineering drawings and the shapes of the objects are defined by the boundary curves without using three-dimensional points on the surfaces. In order to reduce the information on boundary conditions or control points, some systems [3] offer cross-sectional designs of the objects by specifying their profiles. These methods, however, have relatively limited capacity to express free-form objects.

In this paper, we present an algorithm of a designer-oriented specification for surface interpolation, which is mainly based on the information of the boundary curves. This algorithm can be used by the designer for his early design stage because it generates surfaces with large areas. The algorithm is a generalization of the surface interpolation techniques of Coons, to which it applies additional operations for rotating and scaling the boundary curves. The interpolation techniques proposed here were originally developed for and used in animation [6], [7].

This algorithm has been incorporated into a large scale computer for the computer aided design and manufacturing system for plastic molding dies [4].

*Central Research Laboratory, Hitachi, Ltd. Kokubunji, Tokyo, 185, Japan.
**Systems Development Laboratory, Hitachi, Ltd. Kawasaki, Kanagawa 215, Japan.

## 2. Representation of Coons Surface

Sculptured surfaces in engineering have traditionally been used in the field of aircraft, ship and automobile design. They have also been extensively used in the design of consumer products. Usually in these fields, the shapes of the objects are determined through the use of clay or wooden models of the objects. Points on the models' surfaces are digitized to construct mesh networks by which sculptured surfaces are represented.

One of the earliest and best known methods of surface description for sculptured surface design was introduced by Coons. The Coons patch is expressed in the following matrix form.

$$\begin{aligned} S(u, v) = &[b_0(u)\ b_1(u)]\begin{bmatrix} S(0, v) \\ S(1, v) \end{bmatrix} \\ &+ [S(u,0)\ S(u, 1)]\begin{bmatrix} b_0(v) \\ b_1(v) \end{bmatrix} \\ &- [b_0(u)\ b_1(u)]\begin{bmatrix} S(0, 0) & S(0, 1) \\ S(1, 0) & S(1, 1) \end{bmatrix}\begin{bmatrix} b_0(v) \\ b_1(v) \end{bmatrix} \end{aligned} \qquad (2.1)$$

where $b_0(u)$, $b_1(u)$, $b_0(v)$ and $b_1(v)$ are the blending functions and

$$b_0(0)=1, \qquad b_0(1)=0$$
$$b_1(0)=0, \qquad b_1(1)=1 \qquad (2.2)$$

$$b_0(u)+b_1(u)=1,\ b_0(v)+b_1(v)=1.$$

As shown in Fig. 2.1, it is assumed here that parameters $u$ and $v$ vary from 0 to 1 along the relevant boundaries, and $S(u, v)$, $0 \le u, v \le 1$, represents the interior of the surface patch. $S(u, 0)$, $S(1, v)$, $S(u, 1)$ and $S(0, v)$ represent the four known boundary curves. In order to obtain this expression, Forrest [9] outlined a treatment in which the sum of the ruled surface was obtained from linear interpolation between pairs of boundaries.

In order to prepare for our representation, we treat the Coons algorithm as follows:

$$\begin{aligned} S_1(u, v) &= [S(u, 0)-S(0, 0)]+S(0, v) \\ S_2(u, v) &= [S(u, 0)-S(1, 0)]+S(1, v) \\ S_3(u, v) &= [S(u, 1)-S(0, 1)]+S(0, v) \\ S_4(u, v) &= [S(u, 1)-S(1, 1)]+S(1, v) \end{aligned} \qquad (2.3)$$

Each of these equations expresses the surface which is interpolated by moving a boundary curve along one of its adjacent boundary curves. In the first step, $S_1$ and $S_2$ are averaged by using the blending functions $b_0(u)$ and $b_1(u)$, and $S_3$ and $S_4$ are blended along the $u$-direction with $b_0(u)$ and $b_1(u)$. In the second step, the above two results are blended along the $v$-direction. The result can be expressed in the form

$$\begin{aligned} S(u, v) = b_0(v)[b_0(u)S_1(u, v)+b_1(u)S_2(u, v)] \\ +b_1(v)[b_0(u)S_3(u, v)+b_1(u)S_4(u, v)]. \quad (2.4) \end{aligned}$$

(2.1) can be derived by rearranging each term of (2.4). Note here that this formulation has symmetry between $u$ and $v$, since the blending procedure can be exchanged between $u$ and $v$.

## 3. Representation of highly-curved surfaces

The Coons patch is constructed solely in terms of information given on its boundary and certain auxiliary scalar functions of $u$ and $v$. When a highly-curved surface with a large area is generated, if a curve-network on the surface is given, the Coons algorithm can construct the surface as an aggregation of the patches, that is, the shape of the objects has to be composed by a network of topologically rectangular patches.

The designer, however, wishes to deal with the shape of the objects by using characteristic curves on engineering drawings, rather than an aggregation of control points. A surface representation is therefore introduced. This representation is suitable for the conventional design method in which the designer makes no models to digitize points on the surfaces, and generates large patches without using information on interior points or curves of the surfaces.

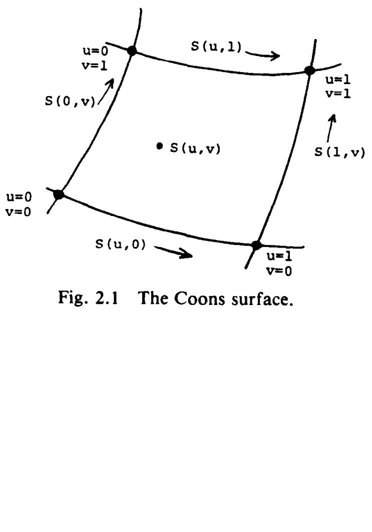


Fig. 2.1 The Coons surface.

## 4. Rotation and scaling operations

### 4.1 Type 1 Rotation Operation

Surfaces such as tori, cylinder or spheres can be easily produced by rotating a curve about an axis, but when the profile is not a circular arc, the above techniques do not work well. Revolutional surfaces use the information on the axis, the central angle and the distance from the center to the points on the surfaces.

When constructing shapes such as the torus shown in Fig. 4.1, in which the characteristic curves are drawn by thick lines, the algorithm is required to generate a highly-curved shape. If the Coons algorithm is applied to these curves, it produces an unwanted shape, such as the one shown in Fig. 4.2. This is because the Coons algorithm uses linear interpolation.

In order to obtain the shape illustrated in Fig. 4.1, it is necessary to add movement of rotation along a characteristic curve (called the guiding curve) to the Coons algorithm.

If we suppose that $S(u, 0)$ is rotated along $S(0, v)$, and that $\partial S(u, v)/\partial u$ and $\partial S(u, v)/\partial v$ are tangent vectors in

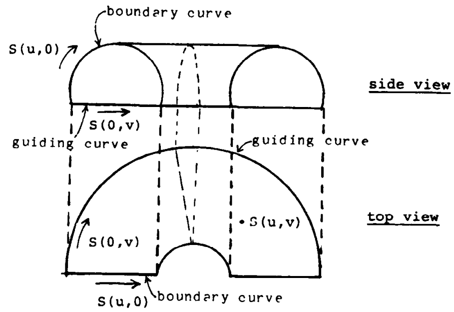


Fig. 4.1 Rotating factor.

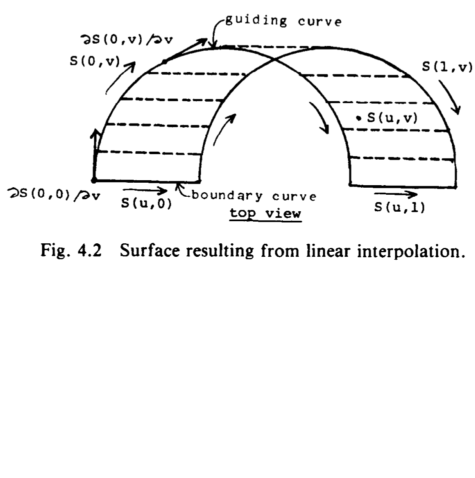


Fig. 4.2 Surface resulting from linear interpolation.

the $u$- and $v$-direction respectively, a surface is obtained by the operation where the boundary curve $S(u, 0)$ is rotated around an axis $N(v)$ (cross product between $\partial S(0, 0)/\partial v$ and $\partial S(0, v)/\partial v$) at a temporary origin $S(0, 0)$ and is displaced to $S(0, v)$. This is expressed as

$$S(u, v)=T_1(v)[S(u, 0)-S(0, 0)]+S(0, v) \qquad (4.1)$$

where

$T_1(v)$ is a 3 by 3 matrix and it is obtained as follows by calculating cos $\theta$ and sin $\theta$ between the two tangent vectors.

$$\begin{bmatrix} n_1^2+c_\theta\cdot(1-n_1^2) & n_1\cdot n_2(1-c_\theta)-n_3\cdot s_\theta & n_3\cdot n_1\cdot(1-c_\theta)+n_2\cdot s_\theta \\ n_1\cdot n_2\cdot(1-c_\theta)+n_3\cdot s_\theta & n_2^2+c_\theta\cdot(1-n_2^2) & n_2\cdot n_3\cdot(1-c_\theta)-n_1\cdot s_\theta \\ n_3\cdot n_1\cdot(1-c_\theta)-n_2\cdot s_\theta & n_2\cdot n_3\cdot(1-c_\theta)+n_1\cdot s_\theta & n_3^2+c_\theta\cdot(1-n_3^2) \end{bmatrix}$$

and,

$$N(v)=\begin{bmatrix} n_1(v) \\ n_2(v) \\ n_3(v) \end{bmatrix}=\partial S(0, 0)/\partial v\times\partial S(0, v)/\partial v/\,|\partial S(0, 0)/\partial v\times\partial S(0, v)/\partial v|$$

$$c_\theta=\cos\theta(v)=\partial S(0, 0)/\partial v\cdot\partial S(0, v)/\partial v/\,|\partial S(0, 0)/\partial v||\partial S(0, v)/\partial v|$$

$$s_\theta=\sin\theta(v)=\partial S(0, 0)/\partial v\times\partial S(0, v)/\partial v/\,|\partial S(0, 0)/\partial v||\partial S(0, v)/\partial v|$$

Thus, the matrix $T_1(v)$ represents rotation along $S(0, v)$. We can obtain a total of eight rotation matrices $M_i$ ($i=1, 8$) by this procedure for every combination of $u$ and $v$ (see Fig. 4.3). If we write the above matrix $M_1$ by representing $T_1(S(0, 0), S(0, v))$, we can express the matrices $M_i$ as

$$\begin{aligned} M_2(v)&=T_1(S(1, 0), S(1, v)),\\ M_3(v)&=T_1(S(0, 1), S(0, v)),\\ M_4(v)&=T_1(S(1, 1), S(1, v)),\\ M_5(v)&=T_1(S(0, 0), S(u, 0)),\\ M_6(v)&=T_1(S(0, 1), S(u, 1)),\\ M_7(v)&=T_1(S(1, 0), S(u, 0)), \end{aligned} \qquad (4.2)$$

and

$$M_8(v)=T_1(S(1, 1), S(u, 1)).$$

### 4.2 Type 2 rotation operation

If the guiding curve is a three-dimensional curve, the axis of revolution is changed as the parameter $u$ or $v$ proceeds. It is difficult to visualize the direction of the axis. If a plane, on which moved curves are fixed, is considered, we suppose that it is easy to catch the movement of the axis, because two dimensional information is presented.

Let us provide another type of rotation. Suppose that $S(u, 0)$ is a characteristic curve and $S(0, v)$ and $S(1, v)$ are the guiding curves. Let us consider a chord vector $K_i(v)$ ($i=1, 2$) connecting two points at $v=vi$ on each of the guiding curves, that is, $K_i(v)=S(1, vi)-S(0, vi)$. A plane $PL_i(v)$ ($i=1, 2$) is determined by $K_i$ and $\partial S(0, vi)/\partial v$. The rotation matrix $T_2(v)$ is obtained by rotating $K_1(v)$ to $K_2(v)$ and then by putting $PL_1(v)$ upon $PL_2(v)$ as shown in Fig. 4.4. This is described by

$$T_2(v)=T_b(v)\cdot T_a(v), \qquad (4.3)$$

where $T_a(v)$ is a rotation matrix determined by turning $K_1(v)$ to $K_2(v)$ around the cross products $K_1(v)$ and $K_2(v)$. $T_b(v)$ is a rotation matrix determined by turning $T_a\cdot PL_1$ to $PL_2$ around $K_2$.

If $K_1=S(1, 0)-S(0, 0)$ and $K_2=S(1, v)-S(0, v)$, $T_2(v)$ is a matrix which rotates $S(u, 0)$ along the guiding curve $S(0, v)$. Matrix $T_2$ can be applied to the same eight combinations as $T_1$ denoted by (4.2).

### 4.3 Scaling operation

Another type of operation is considered here. Suppose that the distance between a pair of the guiding curves $S(0, v)$ and $S(1, v)$ changes along the $v$-direction as depicted in Fig. 4.5. Assuming the boundary curve $S(u, 0)$ is an arc, we can obtain the same shape as $S(u, 0)$ at $v=v_1$ on $S(0, v)$ or $S(1, v)$, if the Coons interpolation

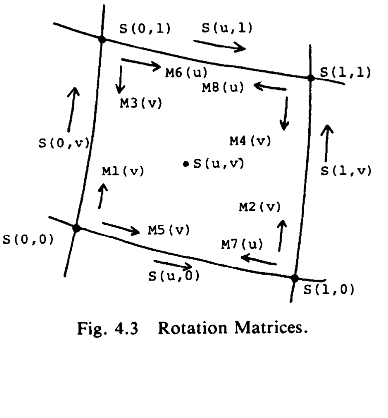


Fig. 4.3 Rotation Matrices.

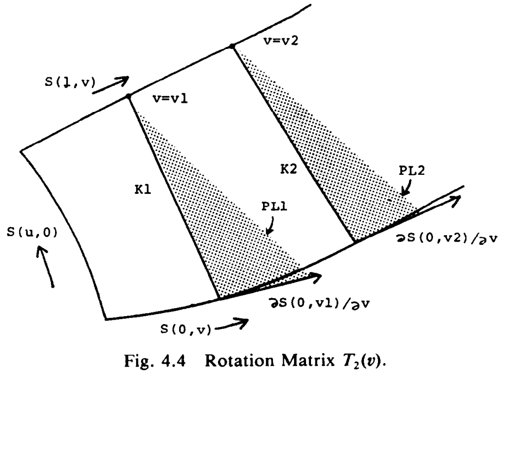


Fig. 4.4 Rotation Matrix $T_2(v)$.

algorithm is applied to this situation. Using a part of the Coons algorithm (2.4), $S(u, 0)-S(0, 0)+S(0, v_1)$ and $S(u, 0)-S(1, 0)+S(1, v_1)$ are blended along the $u$-direction. The result is a relatively flat surface compared with the shape of $S(u, 0)$, if the designer intends to make a part of a cone.

In order to obtain such shapes, the boundary curves should be enlarged or reduced with the ratio between the initial chord length $|S(0, 0)-S(1, 0)|$ and the chord length $|S(0, v_1)-S(1, v_1)|$ at the parameter $v=v_1$.

If we suppose that the boundary curve $S(u, 0)$ is displaced along the $v$-direction as shown in Fig. 4.5, the scaling ratio $r_1$ is determined as

$$r_1(v)=|S(0, 0)-S(1, v)|/|S(0, 0)-S(1, 0)|, \quad (4.4)$$

and this ratio is multiplied to transform the shape of the curve as follows.

$$S(u, v)=r_1(v)[S(u, 0)-S(0, 0)]+S(0, v). \quad (4.5)$$

Similarly we can obtain $r_2(v)$ to $r_4(v)$ as

$$r_2(v)=|S(0, v)-S(1, v)|/|S(0, 1)-S(1, 1)|$$
$$r_3(v)=|S(u, 0)-S(u, 1)|/|S(0, 0)-S(0, 1)|$$
$$r_4(v)=|S(u, 0)-S(u, 1)|/|S(1, 0)-S(1, 1)|$$

As shown above, the boundary curves are rotated and scaled by matrices and ratios. These matrices and ratios are combined into the following matries for convenience of formulation.

$$R_1(v)=r_1(v)M_1(v), \quad R_2(v)=r_1(v)M_2(v)$$
$$R_3(v)=r_3(v)M_3(v), \quad R_4(v)=r_4(v)M_4(v) \quad (4.6)$$
$$R_5(u)=r_5(u)M_5(u), \quad R_6(u)=r_6(u)M_6(u)$$
$$R_7(u)=r_7(u)M_7(u), \quad R_8(u)=r_8(u)M_8(u).$$

The effects of these matrices are selected by the user to control the shape of the surfaces. For example, if the designer does not need to perform any scaling or rotation operations, all of the matrices should be set at a unit matrix I.

Five types of matrices are provided as follows. In Type 1 $R_i$ is a unit matrix. This becomes equivalent to the Coons algorithm. Type 2 has a matrix with $M_i=I$, and then $R_i$ is a unit matrix with diagonal values $r_i$. Type 3 is obtained by substituting $r_i=1$. Type 4 is that where both are effective and $M_i$ is a matrix $T_1$. Type 5 has the same effects as Type 4 except that $M_i$ is a matrix $T_2$.

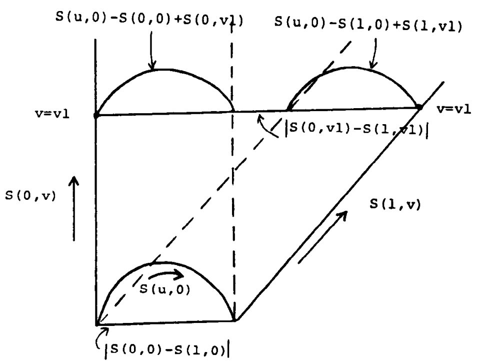


Fig. 4.5 Scaling Factor.

## 5. Direction of the Interpolation

If the designer, however, wishes to obtain surfaces which are expressed by moving the boundary curves along one of the $u$- or $v$-directions, the direction of interpolation should be restricted in the formulation of the interpolation.

In order to represent the restriction of the rotation and scaling factor specified by the designer, we can change the values of the matrices for this purpose. As mentioned in section 2, the Coons equation implies stepwise averaging. Therefore, our algorithm allows the designer to define stepwise blending in each $u$- or $v$-direction by taking some parts of the surface equation terms. Four methods to select the direction are provided as shown in Fig. 5.1.

The combinations for the direction of interpolation (mode) and the type of interpolation (type) are shown in Table 1. The reason some combinations are pro-

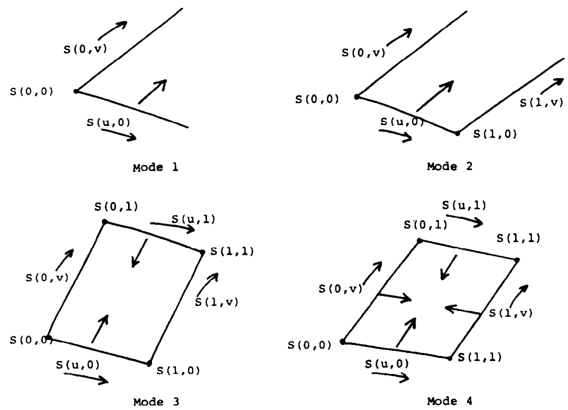


Fig. 5.1 Direction of Interpolation.

Table 1 Combinations of Type and Mode.

| | | MODE 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| TYPE | 1 linear | Yes | Yes | Yes | Yes |
| | 2 ratio | No | Yes | Yes | Yes |
| | 3 rot 1 | Yes | Yes | Yes | Yes |
| | 4 ratio + rot 1 | No | Yes | Yes | Yes |
| | 5 ratio + rot 2 | No | Yes | Yes | Yes |

hibited in Table 1 is that the ratios and the $T_2$ type rotation matrix require a pair of guiding curves. Some examples of the combination for the type and the mode are shown in Fig. 5.2.

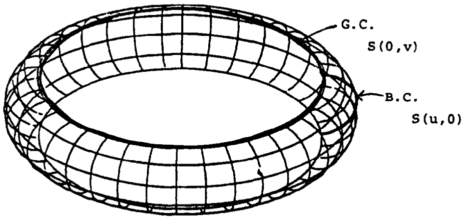


Fig. 5.2a Example: Mode 1 Type 3. G.C.: Guiding Curve, B.C.: Boundary Curve.

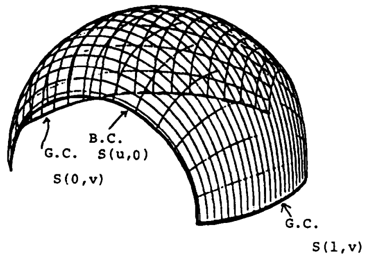


Fig. 5.2c Example: Mode 2 Type 5.

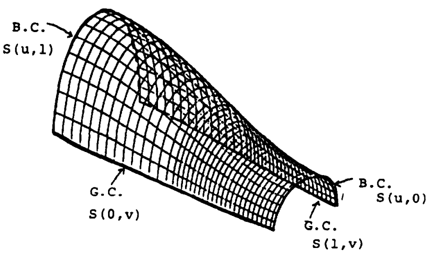


Fig. 5.2b Example: Mode 3 Type 2.

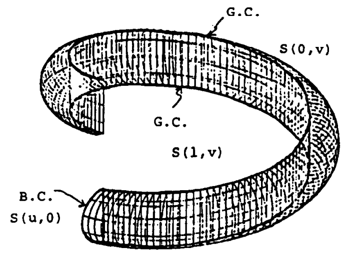


Fig. 5.2d Example: Mode 2 Type 5.

## 6. The algorithm of the interpolation

The effects of the rotation and scaling operations are expressed by matrix multiplication to (2.3). Since the interpolation is performed for the four boundary curves along the two guiding curves which connect both end points of each boundary curve, the following eight surface equations are considered.

$$\begin{aligned}
S_1(u, v) &= R_1(v)[S(u, 0) - S(0, 0)] + S(0, v),\\
S_2(u, v) &= R_2(v)[S(u, 0) - S(1, 0)] + S(1, v),\\
S_3(u, v) &= R_3(v)[S(u, 1) - S(0, 1)] + S(0, v),\\
S_4(u, v) &= R_4(v)[S(u, 1) - S(1, 1)] + S(1, v),\\
S_5(u, v) &= R_5(u)[S(0, v) - S(0, 0)] + S(u, 0),\\
S_6(u, v) &= R_6(u)[S(0, v) - S(0, 1)] + S(u, 1),\\
S_7(u, v) &= R_7(u)[S(1, v) - S(1, 0)] + S(u, 0),\\
S_8(u, v) &= R_8(u)[S(1, v) - S(1, 1)] + S(u, 1),
\end{aligned} \tag{6.1}$$

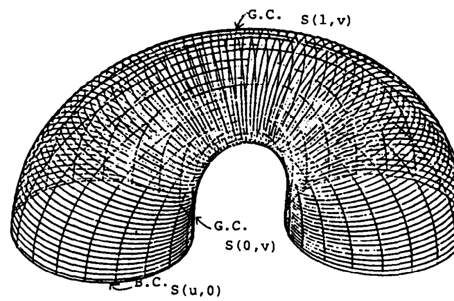


Fig. 5.2e Example: Mode 2 Type 4.

and these eight surfaces are blended along the $u$- or $v$-direction. Two different surfaces are obtained along each direction and these two surfaces are averaged as follows.

$$\begin{aligned}
S(u, v) = {} & 1/2[b_0(v)[b_0(u)S_1(u, v) + b_1(u)S_2(u, v)] + b_1(v)[b_0(u)S_3(u, v) + b_1(u)S_4(u, v)]]\\
& + 1/2[b_0(u)[b_0(v)S_5(u, v) + b_1(v)S_6(u, v)] + b_1(u)[b_0(v)S_7(u, v) + b_1(v)S_8(u, v)]]
\end{aligned} \tag{6.2}$$

This expression is conveniently written by the matrix form

$$\begin{aligned}
S(u, v) = {} & [b_0(u)\ b_1(u)] \begin{bmatrix} A_1(v) & A_3(v) \\ A_2(v) & A_4(v) \end{bmatrix} \begin{bmatrix} S(u, 0) & 0 \\ 0 & S(u, 1) \end{bmatrix} \begin{bmatrix} b_0(v) \\ b_1(v) \end{bmatrix}\\
& + [b_0(u)\ b_1(u)] \begin{bmatrix} S(0, v) & 0 \\ 0 & S(1, v) \end{bmatrix} \begin{bmatrix} A_5(u) & A_6(u) \\ A_7(u) & A_8(u) \end{bmatrix} \begin{bmatrix} b_0(v) \\ b_1(v) \end{bmatrix}
\end{aligned} \tag{6.3}$$

$$-[b_0(u)\ b_1(u)]\begin{bmatrix} C_1(u,v)S(0,0) & C_3(u,v)S(0,1) \\ C_2(u,v)S(1,0) & C_4(u,v)S(1,1) \end{bmatrix}\begin{bmatrix} b_0(v) \\ b_1(v) \end{bmatrix}$$

where

$$A_i(v)=[I+R_i(v)]/2 \qquad (i=1,2,3,4)$$
$$A_i(u)=[I+R_i(u)]/2 \qquad (i=5,6,7,8)$$
$$C_1(u,v)=[R_1(v)+R_5(u)]/2$$
$$C_2(u,v)=[R_2(v)+R_7(u)]/2$$
$$C_3(u,v)=[R_3(v)+R_6(u)]/2$$
$$C_4(u,v)=[R_4(v)+R_8(u)]/2.$$

By substituting $A_i=I$ and $C_i=I$ into (6.3), the Coons algorithm (2.1) can be obtained. The mode is determined as follows. Mode 1 is $S_1(u, v)$ in (6.1). Mode 2 is the first term with $S_1$ and $S_2$ in (6.2). The first line in (6.2) is taken for Mode 3. In the case of Mode 3, the equation is expressed by substituting

$$A_i(v)=R_i(v) \quad (i=1,2,3,4),$$
$$A_i(u)=I \quad (i=5,6,7,8),$$

and

$$C_i(u,v)=R_i(v) \quad (i=1,2,3,4)$$

into (6.3).

## 7. Implementation

The algorithm has been incorporated into a CAD/CAM system for plastic molding dies on a HITAC M-680H large scale computer running a VOS3 operating system. The display system is a Ramtek 9400 and a Tektronix 4114 graphics display.

The system as implemented allows the boundary curves and surfaces to be described by a special purpose language which is similar to APT, and also allows perspective views to be displayed on the screen while calculating cutter paths for numerical control (NC) machines.

Fig. 7.1 shows a perspective view of the characteristic curves of an actual consumer product. The surfaces are generated as shown in Fig. 7.2. There are 46 surfaces in the Figure and the two large surfaces on the side are interpolated by Type 2 rotation and scaling operation. Fig. 7.3 shows generated cutter paths which are calculated in 200 sec. on the HITAC M-680H computer.

The parameter assignment is based on chord-length parametrization. The algorithm allows the shape of the boundary curves to be composed of several curve elements, e.g. like line segments, circular arcs or spline curves. To preserve the knots between the elements of the two curves, which are the characteristic points on the curves, the chord-length parametrization is carried out for each side of a surface before the parameters of

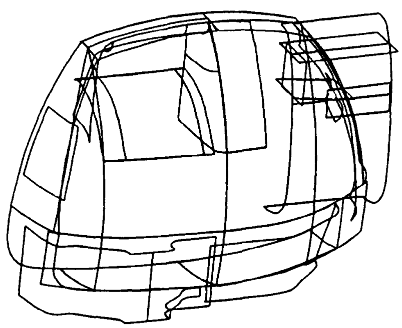

Fig. 7.1 Characteristic Curves.

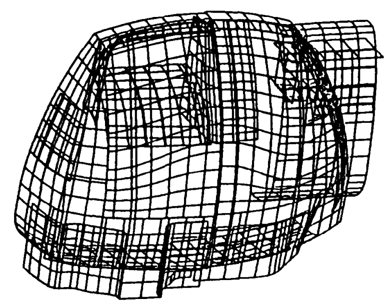

Fig. 7.2 Generated Surfaces.

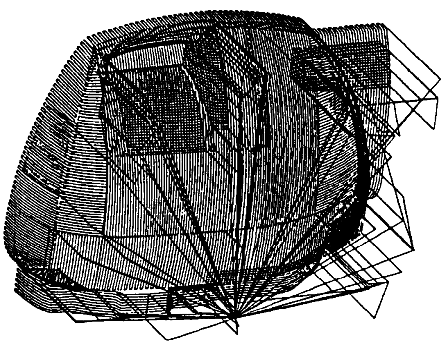

Fig. 7.3 Generated Cutter Path.

both sides are averaged at the corresponding characteristic points on the curves. The choice of parametrization is left to the designer, who can chose to assign either uniform or non-uniform parameter values to the characteristic points in the $u$, $v$ direction.

It is necessary to generate tangent vectors on surfaces when calculating cutter paths or color shaded pictures. It is difficult to obtain analytic derivatives easily from the surface equations. The gradients are calculated numerically using the points on the surface.

## 8. Conclusion

A designer-oriented surface representation method has been developed and implemented. This method is achieved by a generalization of the Coons interpolation technique and the representation of the effects for rotating and scaling boundary curves in the algorithm. Several different surface shapes can be generated in accordance with the designer's specifications, since the choice of the rotating and scaling operations and the interpolation direction is left to the designer. This technique allows relatively complex surfaces to be generated by using characteristic curves on sketches or drawings which are usually specified by the designer. This method has been successfully applied in the production of practical consumer electrical appliances.

## Acknowledgments

The authors would like to acknowledge the advice and encouragement of Dr. Hisashi Horikoshi and Tokinori Kozawa of Hitachi Ltd. We also wish to thank the Computer Graphics Research Group members of Central Research Laboratory, especially Tsuneya Kurihara and Michihiro Shimizu, for their good programming. Thanks are also due to Emiko Oyama for her assistance in the preparation of this paper.

**References**

1. BEZIER, P. E. Numerical Control—Mathematics and Applications, John Wiley and Sons, London (1972).
2. COONS, S. A. Surface for Computer Aided Design of Space Forms, MIT Project MAC TR-**41** (June 1967).
3. FAUX, I. D. and PRATT, M. J. Computational Geometry for Design and Manufacture, Chapter **8**, ELLIS HORWOOD LIMITED, Chichester (1979).
4. YAJIMA, A. and JONISHI, H., MDM-I: A Computer Aided Mold Design and Manufacturing System, *Proc. of Graphics Interface* **'82** (May 1982) 187–193.
5. RIESENFELD, R. F. Applications of B-spline Approximation to Geometric Problems of Computer Aided Design, Ph.D. thesis, Syracuse University (May 1973).
6. MIURA, T., IWATA, J. and TSUDA, J. An Application of Hybrid Curve Generation—Cartoon animation by *Electronic Computers, Spring Joint Compu. Conf. (1967).*
7. YOSHIMURA, S. and TSUDA, J. A Computer Animation Techniques of 3-D Objects with Curved Surfaces, *Proc. of UAIDE* (1971).
8. FERGUSON, J. C. Multivariable Curve Interpolation, *JACM*, **11**, 2 (February 1964), 221–228.
9. FORREST, A. R. On Coons' and other Methods for the Representation of Curved Surfaces, *Computer Graphics and Image Proc.* **1** (1972), 341–359.

(Received April 1, 1985; revised February 9, 1987)