ROCKY MOUNTAIN
JOURNAL OF MATHEMATICS
Volume 14, Number 1, Winter 1984

A REPRESENTATION FOR THE CONSTRUCTIVE DESIGN OF CURVED SURFACES

THOMAS JENSEN

ABSTRACT. Traditionally, there have been two major methods for the representation of complex, curvilinear shapes. Objects are represented by binary trees of set operations upon a restricted set of rigid forms, such as cylinders, spheres, etc., or by an explicit piecewise definition of the boundary by polygons and patches. The former is intuitive for modeling, but lacks generality, particularly in the area of smoothness. The latter, though sufficiently general for representation, is often difficult to model in environments where properties such as symmetry are as important as interpolatory constraints. This paper presents a functional representation, called Constructive Surface Geometry, which unifies both approaches and which possesses advantages of each.

Introduction. Currently, most solid, animation and simulation production modeling systems are not based upon a primitive capable of representing smooth, curved surfaces. [5] There are several reasons for this.

1. It is much more difficult to render smooth shapes than faceted ones. The convex planar polygon has a unique property; its perspective projection is always silhouetted by the projection of its boundary. This is not true of any curved surface, where determination of the silhouette after perspective projection entails the solution of a non-linear equation.

2. The increase in computation required to manipulate curved objects is very large. Calculation of intersections of faceted objects leads to linear systems; calculation of intersections of curved objects requires solution of non-linear systems.

3. While it is well known that any continuous surface can be uniformly approximated to within arbitrary $\varepsilon > 0$ by a network of patches, this says nothing about the ability of a designer to model curved features in a particular system. [3] In fact, designing by interpolatory constraints alone, even in an interactive graphical environment where dynamic rendering of three-dimensional objects is possible, is often difficult. This follows from the necessity of modeling through a two-dimensional interface and either using several orthographic projections, or a single perspective projection. The limitations of human depth perception are all too

apparent in modeling environments developed around the latter, and the overhead incurred with the former by requiring several actions to alter the position of one point, is often unacceptable. [6]

Some systems, such as General Motors' GMSOLID, include a patch primitive. These systems are not limited by the first two problems; that of rendering smooth shapes through the solution of a non-linear equation, or the increase in computation required to manipulate curved objects. For example, homotopy continuation methods are sufficient for intersections in GMSOLID. [2] But the use of the patch primitive can be constrained by the third reason; the ability of a designer to model curved features using interpolatory constraints alone.

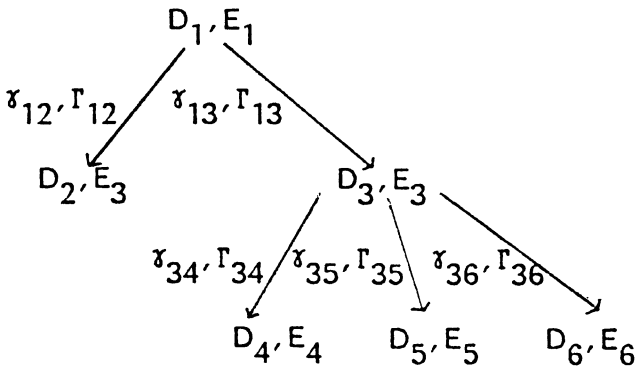
General curvilinear geometry will always be more expensive and complex than piecewise linear representation. The first two problems will always exist. It is only by solving the third problem, that its use can be justified. Only when it is possible to design intuitively and efficiently in a curved domain is the consequent complexity of the data and programs acceptable.

A representation. For designers to effectively manipulate curved features, they must have general and global structures for control as well as the necessarily local tool of manipulation of interpolatory constraints. These general and global structures must provide greater flexibility than the standard affine transformations: rotations, scales and translations. Symmetry should be handled naturally; the designer should be able to uniformly affect an arbitrary continguous portion of the surface of an object. Further, there should be a method by which portions of common curved objects, such as spheres, ellipsoids and cylinders could interact with and constitute portions of more complex objects. Constructive Surface Geometry is proposed as a means of accomplishing these objectives.

In this approach, shapes are represented as sets of parametric surfaces which may overlap, blend together or meet smoothly. The parametric surfaces are traditional winged-edge networks of rectangular and triangular patches. The linkages between them form a union of n-ary trees in which descendence indicates how one surface has been placed, and perhaps blended, into another. More precisely, the branches correspond to two related affine mappings: one which associates the domains of the parent and descendent and one which associates the range of the parent with the graph of the descendent. This method of combination is generalized by the inclusion of blending.

Specifically, let $\{f_i\}$, $i = 1, n$ be a set of functions, each of which maps a piecewise linearly bounded subset of the plane into $\mathbf{R}^3$. Denote the domain of f_i by D_i and the graph in $\mathbf{R}^3$ of f_i by E_i. Let γ_{ij} denote an affine mapping from D_i to D_j and Γ_{ij} denote an affine mapping from E_i

into $\mathbf{R}^3$. Then the representation consists of a function G defined with a set $B = \{D_i, E_i, D_j, E_j, \gamma_{ij}, \Gamma_{ij}\}$, $i \neq j$, which describes the linkages of the tree structures. More intuitively, a particular representation might be thought of in terms of the following graph, where the arrows denote the direction of the affine embeddings.



The function G is defined recursively. Descendents are combined with their parents one level at a time, beginning at the leaves of the tree and proceeding upwards towards the root. The result supersedes the parent in the next iteration and the descendent is deleted. The recursion is complete when only one node remains.

Thus, if $(D_i, E_i, D_j, E_j, \gamma_{ij}, \Gamma_{ij})$ is an element of B, then at some point f_i is replaced by

$$\phi_{ij}(s, t)f_i(s, t) + \theta_{ij}(\gamma_{ij}(s, t))\, \Gamma_{ij}(f_j(\gamma_{ij}(s, t))),$$

where ϕ_{ij} is a univariate blending function based on distance to the boundary of the projection of D_j in D_i, θ_{ij} is a univariate blending function based on distance from the inside to the boundary of D_j, and (s, t) resides in D_i. The blending functions, which are described later, are chosen so that

1. $\phi_{ij} + \theta_{ij} = 1$, for all s and t,
2. ϕ_{ij} is identically one outside of an open set in D_i containing the projection under γ_{ij} of D_j, and
3. θ_{ij} is identically one on a closed set in D_j.

In the particular case of the graph above, f_6 and f_3 would be combined, then f_5 and f_3, followed by f_4 and f_3. Function f_3 would then be blended into f_1 and g would be defined by the final combination of f_2 and f_1. The advantages of this type of representation are the following three reasons.

1. Surfaces, though more general than patch networks, are still defined

as functions. Traversing the domains of the top nodes traverses the complete surface.

2. There is greater generality at boundaries.
3. Assembly and blending of surfaces is natural.

An example. There are, of course, a large number of affine maps and univariate blending schemes. The following simple and specific example shows how the structure can be used in a natural way to define complex, curved geometries. It demonstrates, and the photographs from an experimental implementation in the concluding section allude to the representation's generality and applicability to a large variety of surface construction problems.

In the example we consider the problem of placing some surface, such as a hemisphere or a fin like that swept from the spline in photograph 1, in the midst of another, such as a plane or a rounded box shape. We assume that both domains are starlike; that is, every ray emanating from the centroid of a domain passes through its boundary only once. The graph associated with the set B is then very simply

$$\begin{array}{l} D_1,\ E_1 \\ \big\downarrow \gamma_{12},\ \Gamma_{12} \\ D_2,\ E_2 \end{array}$$

and $G(s, t)$ can be written explicitly as

$$G(s, t) = \phi_{1,2}(s, t)f_1(s, t) + \theta_{1,2}(\gamma_{1,2}(s, t))\Gamma_{1,2}(f_2(\gamma_{1,2}(s, t))).$$

In this example, and in those from the experimental implementation, γ_{ij} is defined by a translation which identifies two points from the two domains D_1 and D_2 and an arbitrary rotation which orients the second domain, (and later the graph of the second function), relative to the first. Thus, $\gamma_{1,2} = ((s, t) - (a, b))R + (c, d)$ where $(s, t), (a, b) \in D_1$, $(c, d) \in D_2$ and

$$R = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}.$$

It follows from substitution that $\gamma_{1,2}(a, b) = (c, d)$. The particular form was chosen because it lends itself naturally to an interactive graphics environment. One can easily imagine a designer orienting one surface relative to another by associating two points, one on each surface and, consequently, one in each domain. By implication, $\Gamma_{1,2}$, the corresponding map in the range which associates the graphs of f_1 and f_2, has the following form:

$$\Gamma_{1,2}(x, y, z) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ n_1 & n_2 & n_3 \end{bmatrix} (x, y, z) + f_1(a, b)$$

where $(n_1, n_2, n_3) = (\partial f_1/\partial s \times \partial f_1/\partial t)/\|(\partial f_1/\partial s \times \partial f_1/\partial t)\| \,|_{(s,t)=(a,b)}$, the unit normal to f_1 at (a, b).

As stated above, $\phi_{1,2}$ and $\theta_{1,2}$ are blending functions. In this case, they are one of the univariate, cubic Hermite basis functions composed with functions measuring distance to the boundary from the exterior in the case of the former, and from the interior in the case of the latter. Specifically, $\theta_{1,2}(u, v) = h_0(\sigma_2(u, v))$, $(u, v) \in D_2$ and $\phi_{1,2}(s, t) = 1 - \theta_{1,2}(\gamma_{1,2}(s, t))$, $(s, t) \in D_1$. σ_2 is defined as follows. Let $\delta D_2 = \{(1 - \alpha)(u_{i,1}, v_{i,1}) + \alpha(u_{i,2}, v_{i,2})\}$, $i = 1, m$ and $0 \leqq \alpha \leqq 1$. For every $(u, v) \in D_2$, the centroid, (u_c, v_c) of D_2 and (u, v) uniquely defines a ray which intersects δD_2 at one point, (u_p, v_p), on the i-th segment. Let $\tau(s, t)$ denote the ratio of the length of the segment (u, v), (u_p, v_p) to that of the segment (u_c, v_c), (u_p, v_p). By Cramer's rule, τ can be written explicitly as

$$\tau(u, v) = 1 - \frac{(u_{i,2} - u_{i,1})(v_c - t) - (v_{i,2} - v_{i,1})(u_c - s)}{(u_{i,2} - u_{i,1})(v_c - v_{i,1}) - (v_{i,2} - v_{i,1})(u_c - u_{i,1})}.$$

Then $\sigma_2(u, v) = \iint \tau(u, v)du\, dv$, where the limits of integration are from $v - \varepsilon$ to $v + \varepsilon$, and from $u - \varepsilon$ to $u + \varepsilon$ respectively. Since τ is a continuous function, σ_2 is C^1. σ_1 is defined similarly. Over all of D_2, σ_2 can be explicitly integrated since it is piecewise linear in s and t.

An implementation: towards an evaluation. The use of tree structures for the representation of geometry is not a new idea; it is at the heart of the constructive solid geometric approach to modeling. [4] There, complex objects are represented as binary trees where the leaves are simple objects, such as spheres and cubes, and the nodes are set operations such as union and intersection. The representation described above is derived from that idea, and is an attempt to bring the advantages of constructive solid geometry to free form surface modeling.

The approach has been implemented, on a limited scale, in an interactive computer graphics environment consisting of a VAX 11–780 computer and an Evans and Sutherland Picture System. In the program, symmetry can be specified at two levels. Initially, symmetric curves can be created by automated reflection about a center line. Later, a primitive surface may be instanced in two symmetric positions simultaneously. Both of those features were used in the definition of the object shown in the series of photographs. The symmetric B-spline in Figure 1 was swept along a linear path to create a fin-shaped open surface. Then, the box shape shown in Figures 3 and 4 was contructed from curves, such as the one shown in

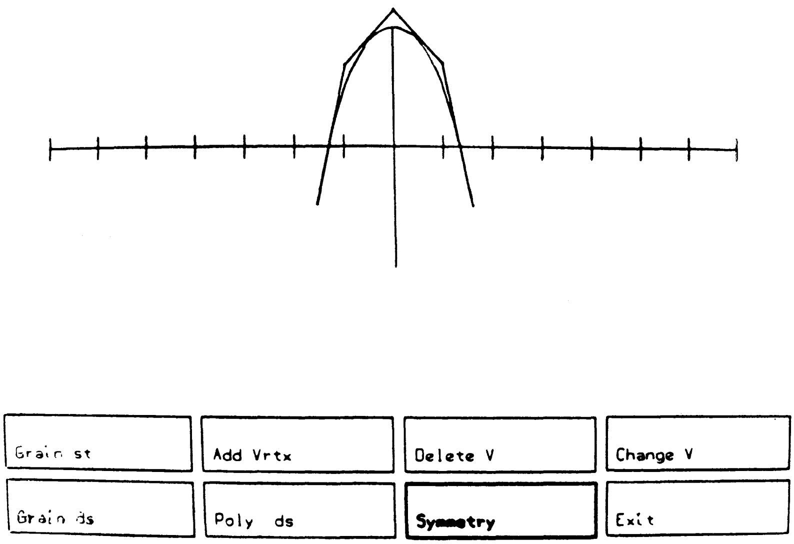


FIGURE 1

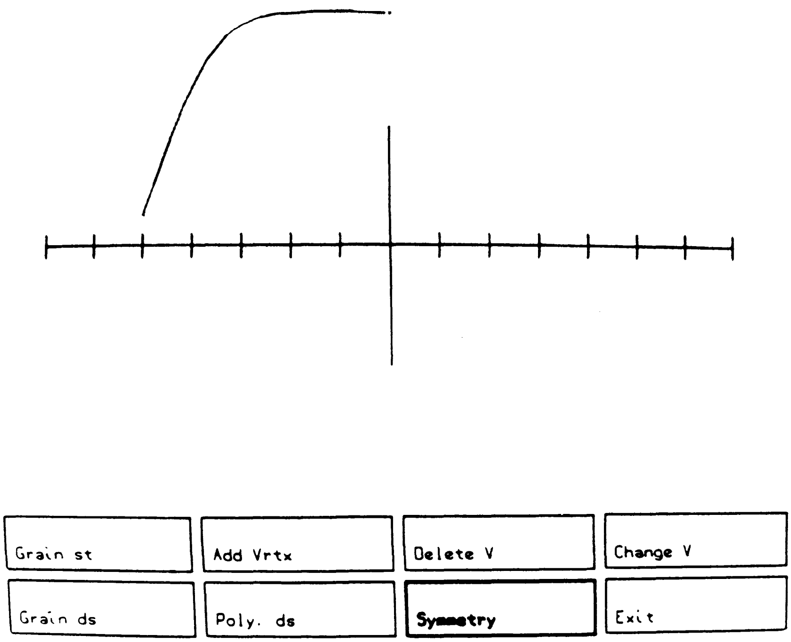


FIGURE 2

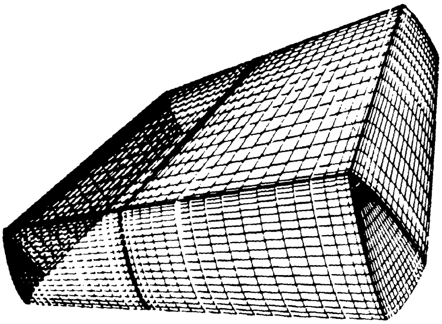


FIGURE 3

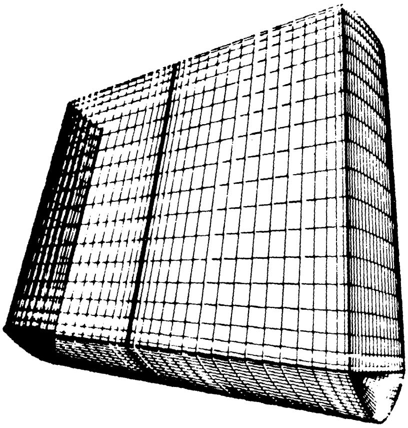


FIGURE 4

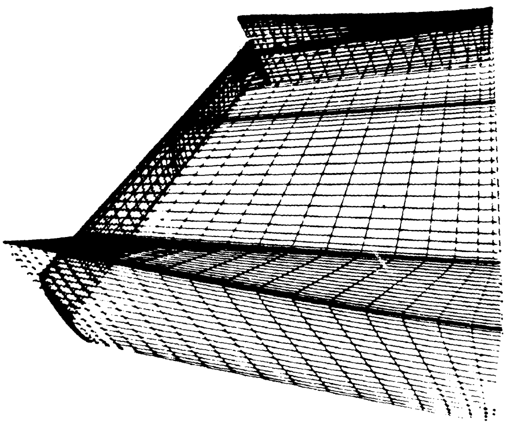


FIGURE 5

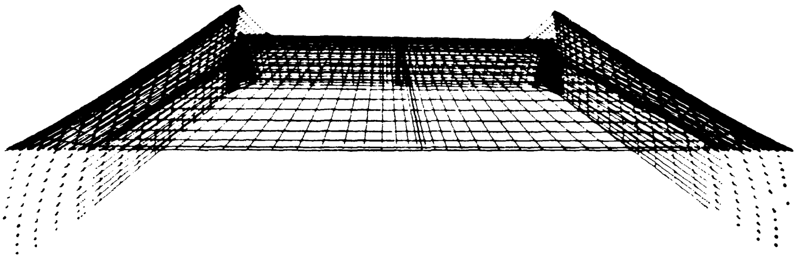
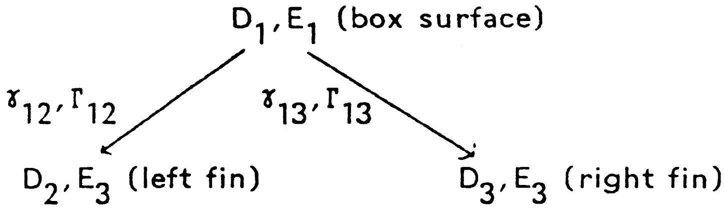


FIGURE 6

Figure 2. Lastly, the fin was instanced and blended into the box surface. The result is represented by the following graph and depicted in Figures 5 and 6.



Finally, it is felt that the value of this approach to a particular problem can be measured in terms of the simplicity of the primitive surfaces that are blended and combined. Further research is planned with the goal of a comprehensive evaluation over a variety of objects.

REFERENCES

1. R. E. Barnhill, *Representation and approximation of surfaces*, Mahtematical Software III, edited by J. Rice, Academic Press, NY, 1977.

2. J. W. Boyse, J. E. Gilchrist, *GMSolid: interactive modeling for design and analysis of solids*, IEEE Computer Graphics, March 1982.

3. P. J. Davis, *Interpolation and Approximation*, Dover, 1975.

4. A. G. Requicha, R. B. Tilove, *Mathematical foundations of constructive solid geometry: general topology of closed regular sets*, Tech. Memo No. 27, Production Automation Project, University of Rochester, New York, March 1978.

5. A. G. Requicha, H. B. Voelcher, *Solid modeling: an Historical summary and contemporary assessment*, IEEE Computer Graphics, March 1982.

6. P. Veenman, *The design of sculptured surfaces using recusive subdivision techniques*, Proceedings of the MIT Conference on CAD/CAM, March 1982.

EVANS AND SUTHERLAND, 580 ARAPEEN DRIVE, SALT LAKE CITY, UT 84108