Computation of global geometric properties of solid objects

H G Timmer and J M Stern

A computational scheme for determining global geometric properties of solid object models is presented. The method operates directly on the boundary representation of the model. The scheme is tested on a number of models produced by an experimental modelling system. Primitive objects combined for the tests are all represented in terms of parametric bicubic patches.

Considerable resources have been devoted to developing geometric modelling systems that seek to represent complex solid objects by combination of simpler objects. A recent survey of such systems is available[1].

In general, the objects are combined using logical operators, which include *union*, *intersection*, and *difference*. To illustrate the impact of these concepts upon geometric modelling, consider the example of two intersecting cylinders shown in Figure 1.

Earlier modelling approaches required an explicit representation of the combined objects. The curves of intersection had to be supplied as inputs such that cylinder A would have two holes in its surface, while cylinder B would consist of two parts, each bounded explicitly by the intersection curve.

A slight relocation of cylinder B would cause the entire input preparation exercise to be repeated. The combinatorial approach allows cylinders A and B to be modelled individually, independent of their eventual combination.

Determination of their intersection is implicit and is ultimately carried out by the computer. Consequently, an exercise such as relocating cylinder B requires the designer to do little more than change a single number. In the explicit approach, the computations to determine the surface area of an object are relatively straightforward, since the domains of integration are rectangular in parametric space. Introduction of the implicit model causes the domain of integration to become irregularly shaped (in general) and therein lies the problem.

During the initial development of such a modelling system, the first application beyond graphics was the computation of global geometric properties such as volume, surface area, centroid, and the moments of inertia. Each of these is expressible as a surface or volume integral whose domain includes the entire object model. The purpose of this paper is to describe an effective computational approach.

The object here is not to provide a detailed description of the data structure, but to give a simplified picture of the boundary representation used.

Douglas Aircraft Company, McDonnell Douglas Corporation, Long Beach, CA, USA

Within the present context, a boundary representation is taken to mean a complete description of the surfaces bounding the solid object. A single surface is defined to be that portion of the boundary that can be mapped into the unit square in parametric space. It should be noted that all surfaces in this prototype system are represented parametrically as opposed to the half-space representation, $F(x, y, z)=0$, used by several modellers.

The consequence of performing any of the combinatorial operations mentioned previously is a new set of bounding surfaces. In general, a surface may be altered by a given operation in a sense that portions of it may be rendered 'inactive', i.e., no longer part of the boundary of the resulting solid object. Figure 2 shows a surface in parametric space that has been so altered.

The parametric representation may be expressed in the usual form.

$$\vec{P} = \vec{P}(u,v) \qquad \begin{matrix} 0 \leqslant u \leqslant 1 \\ 0 \leqslant v \leqslant 1 \end{matrix} \tag{1}$$

As a result of the various operations, the typical surface shown in Figure 2 has only the cross-hatched region remaining as an active portion of the object's bounding surface. Thus, besides the data required to evaluate equation (1), the model also contains representations of the loops bounding the active portion of each surface;

$$\begin{matrix} u = u_i(t) \\ v = v_i(t) \end{matrix} \quad (0 \leqslant t \leqslant 1) \tag{2}$$

where the subscript i identifies the particular loop. Figure 2 also indicates a convention on the direction of parameterization of the loops. The active portion of the surface remains to the left of an observer traversing the curve in the direction of increasing t.

The computational scheme described differs from those employed in most commercial modellers as it operates on the boundary representation directly. Most schemes are based on the modeller's ability to answer the fundamental modelling query of whether a given point is inside or outside the object. The computational schemes are carried out in physical space and are characterized by approximating the object with infinitesimal prisms or similar volumetric elements.

ANALYSIS

Global properties, as opposed to local properties such as curvature, can be expressed as either surface integrals or volume integrals. The volume integrals may be reduced to surface integrals using Gauss's theorem[2]:

$$\iiint_R \vec{\nabla} \cdot \vec{\Phi} \, d\tau = \iint_S \vec{\Phi} \cdot \mathbf{n} \, d\sigma \tag{3}$$

where $\vec{\nabla}$ is the operator $\left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right)$ and $\mathbf{n}$ is the unit sur-

 0010–4485/80/060301–04 $02.00

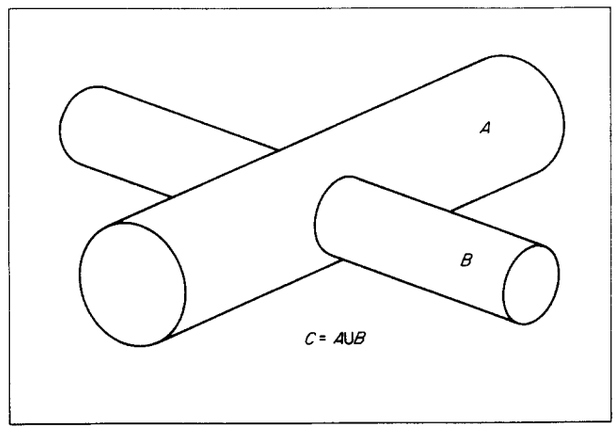

Figure 1. Example of combinatorial modelling

face normal vector, which, by convention, is positive when directed outward. If the property in question is given by:

$$\psi = \int\int\int_R F(x,y,z)d\tau \quad (d\tau = dxdydz) \tag{4}$$

then the vector function $\vec{\Phi}$ must satisfy:

$$\vec{\nabla} \cdot \vec{\Phi} = F \tag{5}$$

The vector $\vec{\Phi}$ is not unique. Consider the case of the moment of inertia about the x-axis, where $F(x,y,z) = y^2 + z^2$. We could choose $\vec{\Phi} = [x(y^2+z^2),0,0]$ or possibly $\vec{\Phi} = (0,y^3/3,z^3/3)$. The property calculation problem has been reduced to evaluating a surface integral of the form:

$$\psi = \int\int_S G(x,y,z)d\sigma \tag{6}$$

If the object model is composed of n surfaces defined parametrically as indicated by equations (1) and (2) and Figure 2, the integral is the sum of integrals over the active portion each surface.

$$\psi = \sum_{i=1}^{n} \int\int_{S_i} G(x,y,z)d\sigma = \sum_{i=1}^{n} \psi_i \tag{7}$$

The integrals ψ_i may be expressed in terms of the parametric variables (u,v) by

$$\begin{aligned}\psi_i &= \int\int_{\tilde{S}_i} G[x(u,v),y(u,v),z(u,v)]\,|J|du dv \\ &= \int\int_{\tilde{S}_i} H(u,v)du dv\end{aligned} \tag{8}$$

where $|J|$ is the Jacobian of transformation and may be expressed as

$$|J| = \left\| \frac{\partial \vec{P}}{\partial u}(u,v) \times \frac{\partial \vec{P}}{\partial v}(u,v) \right\| \tag{9}$$

The domain of integration is the active portion of the surface, as shown in Figure 2. If $\tilde{S}_i$ is regular (eg the unit square), a double Gaussian quadrature is quite effective. The integral is approximated by:

$$\psi_i \cong \frac{1}{4} \sum_{j=1}^{m} \sum_{k=1}^{m} W_j W_k H(a_j,a_k) \tag{10}$$

where the W_j and a_j are the weights and zeroes associated with a Legendre polynomial of order $m+1$.

If the domain of integration is irregular, as in Figure 2, the problem is significantly more difficult. The surface integral may be transformed into a line integral by applying Green's theorem[2], which provides:

$$\int\int_{\tilde{S}_i} \left[\frac{\partial \alpha}{\partial u}(u,v) - \frac{\partial \beta}{\partial v}(u,v)\right] du dv = \oint_{\Gamma_i} [\beta(u,v)du + \alpha(u,v)dv] \tag{11}$$

The path of integration includes the boundary of the active portion of the surface shown in Figure 2 as the loops $\vec{c}_1(t)$ and $\vec{c}_2(t)$. As in the application of Gauss's theorem, the functions α and β must be determined such that

$$\frac{\partial \alpha}{\partial u} - \frac{\partial \beta}{\partial v} = H(u,v) \tag{12}$$

In equation (5), the function $F(x,y,z)$ can be specified independent of the object's geometry so that a determination of $\vec{\Phi}$ can be made by inspection. Unfortunately, the function $H(u,v)$ is generally complex, and the determination of α and β can only be approximately achieved.

It is assumed that the surface defined over $\tilde{S}_i$ is fairly smooth and that the function H is also smooth and well behaved within the region $\tilde{S}_i$. Let the function H be approximated by a polynomial form:

$$H(u,v) = \sum_{i=1}^{M} \sum_{j=1}^{M} A_{ij}\, u^{i-1}\, v^{j-1} \tag{13}$$

Once again, the determination of α and β by equation (12) is not unique, so that it can be assumed that $\beta(u,v) = 0$ without any loss in generality. Thus,

$$\alpha(u,v) = \sum_{i=1}^{M} \sum_{j=1}^{M} \frac{A_{ij}}{i} u^i v^{j-1} \tag{14}$$

Equation (11) becomes

$$\psi_i = \oint_{\Gamma_i} \alpha_i(u,v)dv = \sum_{\ell=1}^{L} \int_0^1 \alpha_i[u_\ell(t),v_\ell(t)] \frac{dv_\ell}{dt} dt \tag{15}$$

where L is the number of loops defining the active portion of $\tilde{S}_i$. In principle, if the loops $\vec{c}_1(t)$ are represented as polynomials, equation (15) may be evaluated exactly since the integrands would be reducible to polynomials in t. However, the order of the resulting polynomials becomes quite large. For example, if cubic polynomials are used all around, the integrand will be a polynomial in t of order 24. Consequently, it was decided to evaluate equation (15) numerically using Gaussian quadrature once again.

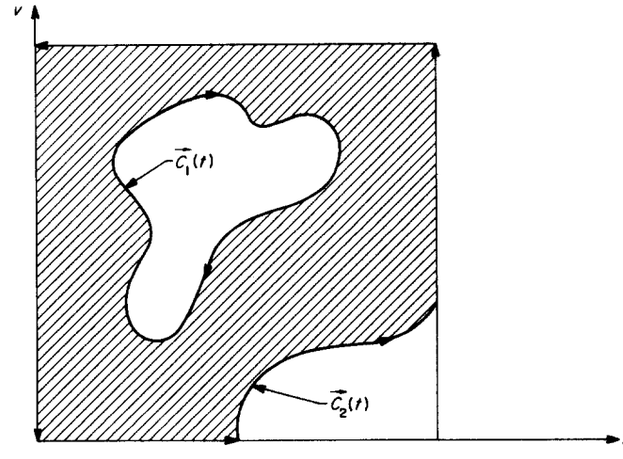

Figure 2. Parametric representation of a typical boundary surface

EXAMPLES

To verify the numerical approach, examples were selected for which the geometric properties could be evaluated in closed form. In all cases, the objects were modelled using rectangular meshes of bicubic patches. Table 1 contains a description of each example solid. The properties calculated for each case are presented in Table 2 with the exact values given in parentheses.

A discussion of the sources of computational inaccuracies may be divided into two categories:

- ordinary computational problems
- geometrical approximations.

The first category includes the quadrature scheme as well as the usual problems of truncation and roundoff that are thoroughly treated in most textbooks on numerical analysis. The second category requires further comment.

The first and probably the most significant cause of inaccuracy is the approximation of the object's bounding surfaces by parametric bicubic polynomials. Since the major consideration of the prototype system implementation is to permit arbitrarily shaped objects, the question of tolerance is considered to be of secondary importance. The representation is exact for the prism, which explains the higher accuracy for examples 1 and 7. Fairly large patches were used to represent the curved surfaces, with the simple rule applied that the tangent vector to any boundary curve can change direction by no more than 45° between corner points.

A second source of error is more subtle and involves the interpolation of the boundary loops stemming from the intersection of two surfaces. The algorithm for computing the intersection between surfaces[3] produces a number of discrete points guaranteed to be within a specified tolerance of the true intersection. Each point (x,y,z) has dual images, one on each surface (u_1,v_1) and (u_2,v_2). The decision was made to interpolate this solution separately for each surface, ie, $u_1(t), v_1(t)$ and $u_2(t), v_2(t)$. While this interpolation (again cubic polynomials) is done to a specified tolerance, it is carried out in parametric space, and the deviation from the true intersection is not controlled.

Table 1. Geometric description of example objects

Example		Mathematical description
1	Unit cube	$(0 \leqslant X, Y, Z \leqslant 1)$
2	(Cylinder)∪(sphere)	$(0 \leqslant X^2 + Z^2 \leqslant .64; -4 \leqslant Y \leqslant 4) \cup (0 \leqslant X^2 + Y^2 + Z^2 \leqslant 1)$
3	Unit sphere	$(0 \leqslant X^2 + Y^2 + Z^2 \leqslant 1)$
4	Torus	$(X = 3\cos u + R\cos v \cos u; y = 3\sin u + R\cos v \sin u; Z = R\sin v)$ $(0 \leqslant u, v \leqslant 2\pi; 0 \leqslant R \leqslant 1)$
5	Ellipsoid	$(0 \leqslant X^2/16 + Y^2/16 + Z^2/25 \leqslant 1)$
6	Cylinder	$(0 \leqslant X^2 + Z^2 \leqslant 64; 0 \leqslant Y \leqslant 1)$
7	[(prism1)−(prism2)] − (prism3)	$(1.5 \leqslant X \leqslant 4.5; 2 \leqslant Y \leqslant 11; 1 \leqslant Z \leqslant 9)$ $-(2.5 \leqslant X \leqslant 3.5; -4 \leqslant Y \leqslant 5; 4.5 \leqslant Z \leqslant 7.5)$ $-(0 \leqslant X \leqslant 6; 0 \leqslant Y \leqslant 9; 0 \leqslant Z \leqslant 6)$
8	Hemi-ellipsoid	$(0 \leqslant X^2/16 + Y^2/16 + Z^2/25 \leqslant 1) -$ $(-5 \leqslant X \leqslant 5; -5 \leqslant Y \leqslant 5; 0 \leqslant Z \leqslant 5)$
9	(Cube)∪(Cylinder)	$(0 \leqslant X, Y, Z \leqslant 4) \cup (0 \leqslant (X-2)^2 + (Z-2)^2 \leqslant 1; -2 \leqslant Y \leqslant 6)$
10	(Cube)−(Cylinder)	$(0 \leqslant X, Y, Z \leqslant 4) - (0 \leqslant (X-2)^2 + (Z-2)^2 \leqslant 1; -2 \leqslant Y \leqslant 6)$

Table 2. Computational results

Example	Surface area	Volume	Centroid	Moments of inertia
1	6(6)	1(1)	$X_c = .5(0.5)$	$I_{xx} = 0.1666667\ (0.1666667)$
			$Y_c = .5(0.5)$	$I_{yy} = 0.1666667\ (0.1666667)$
			$Z_c = .5(0.5)$	$I_{zz} = 0.1666667\ (0.1666667)$
2	45.72(45.74)	16.97(16.99)	$X_c = 0\ (0)$	$I_{xx} = 88.7(88.8)$
			$Y_c = 0\ (0)$	$I_{yy} = 5.84(5.86)$
			$Z_c = 0\ (0)$	$I_{zz} = 88.7(88.8)$
3	12.54(12.57)	4.18(4.19)	$X_c = 0\ (0)$	$I_{xx} = 1.67(1.68)$
			$Y_c = 0\ (0)$	$I_{yy} = 1.67(1.68)$
			$Z_c = 0\ (0)$	$I_{zz} = 1.67(1.68)$
4	118.3(118.4)	59.1(59.2)	$X_c = 0\ (0)$	$I_{xx} = 302.5(303.5)$
			$Y_c = 0\ (0)$	$I_{yy} = 302.5(303.5)$
			$Z_c = 0\ (0)$	$I_{zz} = 576(577)$
5	234.9(235.3)	334(335)	$X_c = 0\ (0)$	$I_{xx} = 2737(2747)$
			$Y_c = 0\ (0)$	$I_{yy} = 2737(2747)$
			$Z_c = 0\ (0)$	$I_{zz} = 2135(2145)$
6	56.52(56.55)	25.11(25.13	$X_c = 0\ (0)$	$I_{xx} = 140.2(140.3)$
			$Y_c = 4\ (4)$	$I_{yy} = 12.54(12.57)$
			$Z_c = 0\ (0)$	$I_{zz} = 140.2(140.3)$
7	185(185)	106.5(106.5)	$X_c = 3\ (3)$	$I_{xx} = 1220.686(1220.685)$
			$Y_c = 7.612675(7.612676)$	$I_{yy} = 555.0378(555.0374)$
			$Z_c = 6.4049290(6.4249296)$	$I_{zz} = 831.399(831.398)$
8	167.6(167.9)	167.1(167.6)	$X_c = 0\ (0)$	$I_{xx} = 781(785)$
			$Y_c = 0\ (0)$	$I_{yy} = 782(785)$
			$Z_c = -1.874(-1.875)$	$I_{zz} = 1067(1072)$
9	121.137(121.133	76.56(76.57)	$X_c = 2\ (2)$	$I_{xx} = 291.0(291.1)$
			$Y_c = 2.0007(2)$	$I_{yy} = 176.96(176.95)$
			$Z_c = 1.9999(2)$	$I_{zz} = 291.0(291.1)$
10	114.86(114.85)	51.44(51.43)	$X_c = 2\ (2)$	$I_{xx} = 150.67(150.77)$
			$Y_c = 2.001(2)$	$I_{yy} = 164.37(164.38)$
			$Z_c = 2.0002(2)$	$I_{zz} = 150.70(150.77)$

Finally, one must consider the approximation that is actually the basis of the present algorithm – the approximation of the integrand by a bipolynomial function. The software is coded so that the order of the approximating polynomials can easily be changed. For the relatively simple examples considered, quintic polynomials appear to provide sufficient accuracy. However, if significant accuracy problems are encountered in future applications, the sensitivity of the solution to the order of the integrand approximations will be among the first things to be examined.

REFERENCES

1 **Baer, A, Eastman, C and Henrion, M** 'Geometric modelling: a survey' *Comput. Aided Des.* Vol 11 No 5 (September 1979) pp 253–272

2 **Kaplan, W** *Advanced calculus* Addison-Wesley Publishing Co., Reading, MA, USA (1956)

3 **Timmer, H** 'A solution to the surface intersection problem' Douglas Aircraft Company Report No MDC-J7789 (November 1977)