Communications

Generation of Configuration Space Obstacles: The Case of a Moving Sphere

CHANDERJIT BAJAJ, MEMBER, IEEE, AND MYUNG-SOO KIM, STUDENT MEMBER, IEEE

Abstract—**Algebraic algorithms are presented for generating the boundary of configuration space obstacles arising from the motion of a sphere among obstacles. The boundaries of the obstacles are given by patches of algebraic surfaces.**

I. Introduction

Using configuration space (C space) to plan motion for a single rigid object among physical obstacles reduces the problem to planning motion for a mathematical point among "grown" configuration space obstacles (the points in C space which correspond to the object overlapping one or more obstacles), Udupa [26], Lozano-Perez and Wesley [18], Lozano-Perez [17]. The C space for the full six degrees of freedom motion is six-dimensional, Canny [10], Donald [13]; however, in the special case of a moving sphere, the C space is three-dimensional. In this correspondence we consider the C-space obstacle generation for a moving sphere and nonconvex obstacles, where each obstacle is bounded by patches of algebraic surfaces. Bajaj and Kim [7] considered the C-space obstacle generation for a translatory motion of a convex object among convex obstacles with algebraic surface boundaries and for nonconvex planar object and obstacles with algebraic curve boundaries, Bajaj and Kim [8]. Most of the essential techniques in the present work come from the results of these two papers. This correspondence thus treats a special case of the extension of Bajaj and Kim [7] to the general case of nonconvex moving objects and obstacles. Bajaj and Kim [9] describe a curvature-dependent hierarchical polyhedral approximation of convex C-space obstacles and its application to compliant motion planning.

The C-space obstacle for a moving sphere is the same as constant radius offsetting of the obstacle. Offsetting as one of the more important operations in geometric modeling because of immediate application in NC machining, has been considered by many authors recently. Farouki [14] outlines exact offset procedures for convex polyhedra, convex solids of revolution, and convex solids of linear extrusion. He also describes algorithms for approximating the offsets of general piecewise parametric surfaces by networks of bicubic patches, where the surfaces are restricted to be smooth on each patch and across adjacent pathes [15]. Tiller and Hanson [25] present an offset capability for planar profiles. Rossignac and Requicha [24] describe offsetting operations for solids represented in the CSG/boundary representation dual scheme where objects are constructed from primitive solids which are natural quadrics (degree two surfaces). This correspondence characterizes the offsetting problem for arbitrary algebraic surfaces and provides an algebraic algorithm for its computations. This algorithm is based on such operations as computing resultants of polynomials, representing surface patches unambiguosly, intersecting two algebraic surfaces, and detecting self-intersections of algebraic surfaces. The efficiency of these operations, however, is quite limited for very high degree algebraic surfaces, a status quo also of geometric modeling.

The main contributions of this work are as follows. In Section III we state that the boundary of C-space obstacles for a moving sphere and nonconvex obstacles is a subset of the convolution of the sphere and the obstacle boundary. In Section IV we give algebraic algorithms to generate the C-space obstacles boundary for nonconvex obstacles. The obstacles are represented by a general algebraic boundary representation (B-rep) model discussed in Section II. Crucial too here is the internal representation of curves and surfaces, i.e., whether they are parametrically or implicitly defined.[1] We present algorithms for both these internal representations. In Section V we consider simple obstacles like solids of revolution, solids of extrusion, or polyhedra, and suggest more efficient algorithms than the general case. Some analytic methods are also considered because of their simplicity in some special cases. Further, we consider relations to blending surfaces in Section VI and summarize and discuss possible extensions of this work in Section VII.

II. Solid Model

In a general boundary representation, an obstacle with algebraic boundary surfaces consists of a list of *peels*. An obstacle may have internal holes and peels which correspond to them are termed "hole" peels. Each peel in turn consists of the following:

1) A finite set of vertices usually specified by Cartesian coordinates.
2) A finite set of directed edges, where each edge is incident to two vertices. (Typically, an edge is specified by the intersection of two faces, one on the left and one on the right. Here left and right are defined relative to the edge direction as seen from the exterior of the object. Further, an interior point is also provided on each edge which helps remove any geometric ambiguity in the representation for high degree algebraic curves, Requicha [21].)
3) A finite set of faces, where each face is bounded by a single oriented cycle of edges. Each face also has a surface equation, represented either in implicit or in parametric form. The surface equation has been chosen such that the gradient vector points to the exterior of the object.

In addition, edge and face adjacency information is provided. Additional conventional assumptions are also made, e.g., edges and faces are nonsingular, two distinct faces intersect only in edges, an auxiliary surface is specifed for each edge where adjacent faces meet tangentially, etc. The object and obstacles that we consider are solids and are assumed to enclose nonzero finite volume. Hence nonregularities such as dangling edges and dangling faces which depending on one's viewpoint enclose zero or infinite volume, are not permitted. The C-space obstacles that we construct are also regularized in this fashion and assumed to be solids enclosing nonzero finite volume. However, the C-space obstacles are at times allowed to have faces with singular points and singular curve boundary edges.

III. C-Space Obstacles and Convolutions

Let A be a moving object whose boundary is a sphere S_r^2 of radius r with its center as the reference point and B be a fixed obstacle in the

Manuscript recieved May 1, 1986; revised August 20, 1987. This work was supported in part by the National Science Foundation under Grant MIP-85 21356 and in part by a David Ross Fellowship.

The authors are with the Department of Computer Science, Purdue University, West Lafayette, IN 47907.

IEEE Log Number 8718078.

[1] A unit sphere is implicitly given as $x^2 + y^2 + z^2 - 1 = 0$ and in rational parametric form as $x = (1 - s^2 - t^2)/(1 + s^2 + t^2)$, $y = 2s/(1 + s^2 + t^2)$, and $z = 2t/(1 + s^2 + t^2)$.

0882-4967/88/0200-0094$01.00 © 1988 IEEE

three-dimensional real Euclidean space R^3. B is modeled by the above boundary representation. We denote $Bdr(B)$ as the boundary of B. Since A does not change its shape by rotations, the configuration space is also three-dimensional. We make the following definitions. 1) $CO(A, B)$ = C-space obstacle due to A and $B = \{\bar{p} \in R^3 | A_{\bar{p}} \cap B \neq \emptyset\}$, where $A_{\bar{p}} = \{\bar{p} + q | q \in A\}$. 2) Convolution $(S_r^2, Bdr(B))$ = convolution of S_r^2 and $Bdr(B) = \{p + r\cdot n_p | n_p$ is a unit outward normal of B at $p \in Bdr(B)\}$. We now note the following.

Theorem 1: 1) $Bdr(CO(A, B)) \subset$ conv $(S_r^2, Bdr(B))$. 2) For convex B, we have $Bdr(CO(A, B))$ = conv $(S_r^2, Bdr(B))$.

Proof: See Bajaj and Kim [7].

This may then suggest a natural method for handling nonconvex obstacles. One first obtains a convex decomposition consisting of union of convex pieces and then generates the C-space obstacle as the union of C-space obstacles for convex obstacles. Such convex decompositions are possible for polyhedral obstacles, see Chazelle [11]. However, not all obstacles with algebraic curve boundaries permit decompositions consisting of the union of convex pieces. For example, a complete toroidal surface cannot be decomposed into the union of convex pieces. To obtain convex decomposition of general curved solid objects (say in terms of union, intersection, and difference) is a difficult and as yet unsolved problem (see Requicha and Voelcker [22]). Hence alternate methods of computing C-space obstacles for nonconvex obstacles become important. The method we suggest here deals with nonconvex obstacles directly.

For nonconvex obstacles with algebraic surface boundaries, the differences between $Bdr(CO(S_r^2, B))$ and conv $(S_r^2, Bdr(B))$ are various kinds of vertices, curve segments, and surface patches. There are primarily five classes of redundancies or singularities that may arise: a) dangling faces, b) isolated vertices and edges, c) coincident faces, d) intersecting faces, and e) self-intersecting faces and faces with singular points or singular curves. We first generate conv $(S_r^2, Bdr(B))$ complete with redundancies. Then on systematically removing redundancies we obtain the C-space obstacle boundary. The following theorem helps characterize this procedure.

Theorem 2: Let O-envelope = conv $(S_r^2, Bdr(B)) \sim \{\bar{p} | A\bar{p}$ and B intersect in an interior point$\}$, where the set difference $P \sim Q = \{p \in R^3 | p \in P$ and $p \notin Q\}$ for P and $Q \subset R^3$. Then O-envelope $\sim Bdr(CO(S_r^2, B))$ is the set of all the isolated edges and vertices, and coincident faces of O-envelope.

Proof: The proof is similar to Bajaj and Kim [8].

IV. Generating the Boundary of C-Space Obstacles

Let $p \in Bdr(B)$ be a vertex, $E \subset Bdr(B)$ be an edge, and $F \subset Bdr(B)$ be a face. One can show that conv $(S_r^2, Bdr(B)) = (\cup_{F\in\Gamma_1}$ conv $(S_r^2, F)) \cup (\cup_{E\in\Gamma_2}$ conv $(S_r^2, E)) \cup (\cup_{p\in\Gamma_3}$ conv $(S_r^2, p))$, where Γ_1 is the set of all faces of $Bdr(B)$, Γ_2 is the set of all edges of $Bdr(B)$, Γ_3 is the set of all vertices of $Bdr(B)$, and conv (S_r^2, K) = conv S_r^2 and $K = \{p + r\cdot n_p | n_p$ is a unit outward normal of B at $p \in K\}$, where $K = F, E$, or p. One can use Section IV-A to compute the convolution faces conv (S_r^2, K). In Section IV-B we consider how to represent each edge of the convolution face conv (S_r^2, K) as an intersection curve of conv (S_r^2, K) and a transversally intersecting auxiliary surface. In Section IV-C we consider how to compute each vertex of the convolution face conv (S_r^2, K). In Section IV-D we consider how to construct the correct topology of conv $(S_r^2, Bdr(B))$, and in Section IV-E we indicate how to remove redundancies of conv $(S_r^2, Bdr(B))$ to get the C-space obstacle boundary.

In the following we consider both the implicit and rational parametric representation of surface patches. Not all algebraic curves and surfaces have rational parametrization, see Walker [28]. For the class of *rational* algebraic curves and surfaces, algebraic algorithms also exist for converting between the implicit and parametric represenations. However, their efficiency are limited to curves and surfaces of low degree, see Abhyankar and Bajaj [3]–[5] and Bajaj [6].

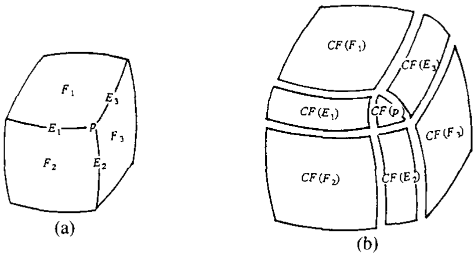

Fig. 1. (a) Faces, edges, and vertices. (b) Convolution faces.

A. Generating Convolution Faces

In this section, we consider how to generate the convolution faces conv (S_r^2, F), conv (S_r^2, E), and conv (S_r^2, p). See Fig. 1 for the generation of convolution faces, where CF(K) means the convolution face generated by $K = F, E$, or p.

Generating Convolutions (S_r^2, F): For each point p on a face F, there is a unique tangent plane L_p at p. The unit outward normal of the plane L_p is the unit outward normal of B at $p \in F$. We can use Theorem 3, and Corollary 1 to generate conv (S_r^2, F). Theorem 3 is useful for the case of F being an implicitly defined algebraic surface, and Corollary 1 is useful when F is parametric

Theorem 3: Let $F \subset Bdr(B)$ be a patch of an algebraic surface $f = 0$ with gradient ∇f. Then conv $(S_r^2, F) = \{\bar{p} = (\bar{x}, \bar{y}, \bar{z}) = p + q = (x + \alpha, y + \beta, z + \gamma)\}$ such that

$$f(x, y, z) = 0 \text{ and } p = (x, y, z) \in F \tag{1}$$

$$\alpha^2 + \beta^2 + \gamma^2 = r^2 \text{ and } q = (\alpha, \beta, \gamma) \in S_r^2 \tag{2}$$

$$\nabla f \times (\alpha, \beta, \gamma) = 0 \tag{3}$$

$$\nabla f \cdot (\alpha, \beta, \gamma) > 0. \tag{4}$$

Proof: Since (3), (4) imply ∇f and (α, β, γ) are in the same direction, (3), (4) are equivalent to the outward normal direction of B at p to be the same as that of S_r^2 at q.

We use Theorem 3 as follows. First substitute $x = \bar{x} - \alpha$, $y = \bar{y} - \beta$, and $z = \bar{z} - \gamma$ in (1) and (3). Then one can obtain the implicit algebraic equation of the conv (S_r^2, F) in terms of $\bar{x}$, $\bar{y}$, and $\bar{z}$ by eliminating α, β, and γ from (1)–(3). The vector equation $\nabla f \times (\alpha, \beta, \gamma) = 0$ gives three scalar equations. Since one of these equations is redundant, we can have two independent scalar equations from (3). Hence, from (1)–(3), we have four equations and we eliminate three variables α, β, γ to get an implicit equation in terms of $\bar{x}$, $\bar{y}$, $\bar{z}$. See Macaulay [19] and van der Waerden [27] for general formulas in elimination.

Corollary 1: Let $F \subset Bdr(B)$ be a parametric surface patch $F(u, v) = (x(u, v), y(u, v), z(u, v))$ with gradient $F_u \times F_v$. Then conv $(S_r^2, F) = \{\bar{p} = (\bar{x}, \bar{y}, \bar{z}) = p + q = (x(u, v) + \alpha, y(u, v) + \beta, z(u, v) + \gamma)\}$ such that

$$p = (x(u, v), y(u, v), z(u, v)) \in F \tag{5}$$

$$\alpha^2 + \beta^2 + \gamma^2 = r^2 \text{ and } q = (\alpha, \beta, \gamma) \in S_r^2 \tag{6}$$

$$(F_u \times F_v) \times (\alpha, \beta, \gamma) = 0 \tag{7}$$

$$(F_u \times F_v) \cdot (\alpha, \beta, \gamma) > 0. \tag{8}$$

First substitute $\alpha = \bar{x} - x(u, v)$, $\beta = \bar{y} - y(u, v)$, and $\gamma = \bar{z} - z(u, v)$ in (6) and (7). Then one can obtain the implicit algebraic equation of conv (S_r^2, F) in terms of $\bar{x}$, $\bar{y}$, and $\bar{z}$ by eliminating u and v from (5)–(7). Since (7) gives two independent scalar equations, we have three equations and eliminate two variables u, v to get a single implicit equation.

Generating Convolution (S_r^2, E): By subdividing an edge E if necessary, we may assume that the inner angles along E between two adjacent faces of E are either a) $> \pi$ (concave), b) $= \pi$ (tangential) or c) $< \pi$ (convex). In the case of a) or b), we contend that there is no unit outward normal of B on E and conv $(S_r^2, E) = \varnothing$. Because of the gap generated by this empty convolution face due to the concave edge E, the convolution faces generated by the two adjacent faces of E may have dangling subfaces. In the case of c), each point $p \in E$ determines two extreme unit outwards normals n_p and $\hat{n}_p$ which are unit outward normals of adjacent faces of E at p. All the unit vectors between these two extreme normals are defined as unit outward normals of B at p. In the following we assume E is of type c). We can use Theorem 4 to generate conv (S_r^2, E) when E is defined by the intersection of two implicit algebraic surfaces. The other combinations of implicit and parametric surfaces defining E have similar results as easy corollaries of Theorem 4.

Theorem 4: Let $E \subset Bdr(B)$ be the common edge of two faces F and $\hat{F}$, where F and $\hat{F} \subset Bdr(B)$ are patches of algebraic surfaces $f = 0$ with gradients ∇f and $\hat{f} = 0$ with gradients $\nabla \hat{f}$. Then conv (S_r^2, E) is the set of points $\bar{p} = (\bar{x}, \bar{y}, \bar{z}) = p + q = (x + \alpha, y + \beta, z + \gamma)$ such that

$$f(x, y, z) = \hat{f}(x, y, z) = 0 \text{ and } p = (x, y, z) \in E \tag{9}$$

$$\alpha^2 + \beta^2 + \gamma^2 = r^2 \text{ and } q = (\alpha, \beta, \gamma) \in S_r^2 \tag{10}$$

$$(\alpha, \beta, \gamma) \cdot (\nabla f \times \nabla \hat{f}) = 0. \tag{11}$$

Proof: Equation (11) is equivalent to an outward normal direction of B at p to be the same as one of the outward normal directions of S_r^2 at q.

Generating Convolution (S_r^2, p)*:* For a vertex p, suppose that there are k edges $E_1, E_2, \cdots, E_k$ adjacent to p (ordered in a counterclockwise direction). Further, suppose $E_{i_1}, E_{i_2}, \cdots, E_{i_{k'}}$ $(k' \le k)$ is the subsequence of all the convex edges. Let e_{i_j} $(1 \le j \le k')$ be the tangent direction of E_{i_j} at p when E_{i_j} is interpreted as a space curve emanating from p. Further, let l_{i_j} be the half line emanating from p to the direction e_{i_j}. Then the point p and the half lines $l_{i_1}, \cdots, l_{i_{k'}}$ determine an infinite pyramid with apex p which is denoted by $\Lambda(p)$. This pyramid locally supports the object S at p. Further, it is either a) concave, b) flat, or c) convex at p. In the case of a) or b), we contend that there is no unit outward normal of B at-p and conv $(S_r^2, p) = \varnothing$. Because of the gap generated by this empty convolution face due to a concave vertex, the convolution faces generated by the adjacent faces and edges of p may have dangling subfaces. In the case of c), the triangular faces of the pyramid $\Lambda(p)$ determine extreme unit outward normals $n_{i_1}, n_{i_2}, \cdots, n_{i_{k'}}$ on S^2. Let γ_{i_j} be the geodesic arc on S^2 connecting n_{i_j} and $n_{i_{j+1}}$, where $i_{k'+1} = i_1$. Then the convex region on S^2 bounded by the closed path $\gamma_{i_1} \rightarrow \gamma_{i_2} \rightarrow \cdots \rightarrow \gamma_{i_{k'}}$ is the set of all unit outward normals of B at the vertex p. Convolution (S_r^2, p) is a convex patch on the translated sphere $(S_r^2)_p$ which is bounded by the closed path $\bar{\gamma}_{i_1} \rightarrow \bar{\gamma}_{i_2} \rightarrow \cdots \rightarrow \bar{\gamma}_{i_{k'}}$ of geodesic arcs $\bar{\gamma}_{i_j}$ connecting the points $p + r \cdot n_{i_j}$ and $p + r \cdot n_{i_{j+1}}$ on $(S_r^2)_p$.

B. Generating Convolution Edges

In this section, we consider how to generate the boundary edges of nonempty convolution faces.

Edges of Convolution (S_r^2, F)*:* Each edge of the convolution face conv (S_r^2, F) is either a) adjacent to an empty convolution face conv (S_r^2, E) if E is concave or tangential, or b) adjacent to a nonempty convolution face conv (S_r^2, E) if E is convex, for some adjacent edge E of the face F. In all these cases, we have difficulties in computing the boundary edges of conv (S_r^2, F) since conv (S_r^2, F) is either adjacent to empty convolution faces or tangent to adjacent convolution faces. Theorem 5 is applicable when the surface path F is an implicitly defined algebraic surface. Further, when F is a parametric surface, we may implicitize this surface to use the following theorem, see Bajaj [6] for a survey of implicitization techniques.

Theorem 5: Let F be a face and E be an edge of F. Suppose E is the common edge of two surface patches F and $\hat{F}$, where F is a patch of an algebraic surface $f = 0$ with gradient ∇f, and $\hat{F}$ is a patch of an algebraic surface $\hat{f} = 0$ with gradient $\nabla \hat{f}$. Then A) the common convolution edge $\bar{E}$ of the convolution faces conv (S_r^2, F) and conv (S_r^2, E) is the set $\{\bar{p} = (\bar{x}, \bar{y}, \bar{z}) = p + q = (x + \alpha, y + \beta, z + \gamma)\}$ such that

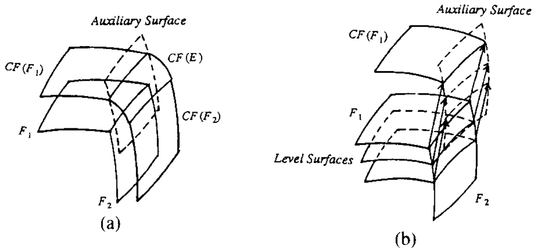

Fig. 2. (a) Auxiliary surface for convolution edge. (b) Relation with level surfaces.

$$f(x, y, z) = 0 \text{ and } p = (x, y, z) \in F \tag{12}$$

$$\hat{f}(x, y, z) = 0 \text{ and } p = (x, y, z) \in \hat{F} \tag{13}$$

$$\alpha^2 + \beta^2 + \gamma^2 = r^2 \text{ and } q = (\alpha, \beta, \gamma) \in S_r^2 \tag{14}$$

$$\nabla f \times (\alpha, \beta, \gamma) = 0 \tag{15}$$

$$\nabla f \cdot (\alpha, \beta, \gamma) > 0; \tag{16}$$

B) the surface patch defined by (12) and (14)–(16) and the surface patch defined by (13)–(16) intersect along the convolution edge $\bar{E}$.

Proof: A) The surface patch defined by (12) and (14)–(16) is the face conv (S_r^2, F) and all its boundary edges and vertices. Since (12), (13) restrict the set of points p to the edge E, (12)–(16) define the convolution edge $\bar{E}$. B) Since $\bar{E}$ is the common solution of (12)–(16) $\bar{E}$ is the common edge of the surface patch defined by (12) and (14)–(16) and the surface patch defined by (13)–(16).

For each point $p \in \hat{F}$, $f(p) = c$ for some level c and the point $\bar{p}$ defined by (13)–(16) is the translation of p by r along the outward normal direction $\nabla f(p)$ of the level surface $f = c$. See Fig. 2, where this surface is shown as a dotted surface patch. When the variation of the vector field $\nabla f / \|\nabla f\|$ in the neighborhood of E is relatively small compared with the differences of normal directions of F and $\hat{F}$ along E, the surface defined by (12) and (14)–(16) and the surface defined by (13)–(16) intersect transversally along $\bar{E}$.

When adjacent convolution faces meet tangentially to each other, computation of the intersecting edge is unstable. Auxiliary surfaces need to be determined which intersect transversally with the convolution surfaces and thereby define boundary curves of the convolution faces. For two surfaces defined implicitly by $h(x, y, z) = 0$ and $\hat{h}(x, y, z) = 0$ which meet tangentially along the curve C, an auxiliary surface which intersects h and $\hat{h}$ transversally may also be obtained by considering surfaces $k = \alpha h + \beta \hat{h} = 0$ where α and β are arbitrary polynominals in three variables x, y, and z. These additional surfaces k also intersect both h and $\hat{h}$ along the curve C and are said to belong to the ideal of the curve C. For suitable α and β auxiliary surfaces which meet h and $\hat{h}$ transversally may be constructed.

For our special case of offset surfaces we may use Theorem 5 to generate an auxiliary surface patch which intersects with conv (S_r^2, F) transversally.

Edges of Convolution (S_r^2, F)*:* When E is a convex edge, each boundary edge of conv (S_r^2, E) is either a) a boundary edge of conv (S_r^2, F) which has been considered above, or b) a geodesic arc γ on the translated sphere $(S_r^2)_p$ connecting two points $p + r \cdot n_p$ and $p + r \cdot \hat{n}_p$, where n_p and $\hat{n}_p$ are unit gradients of adjacent faces of E at p.

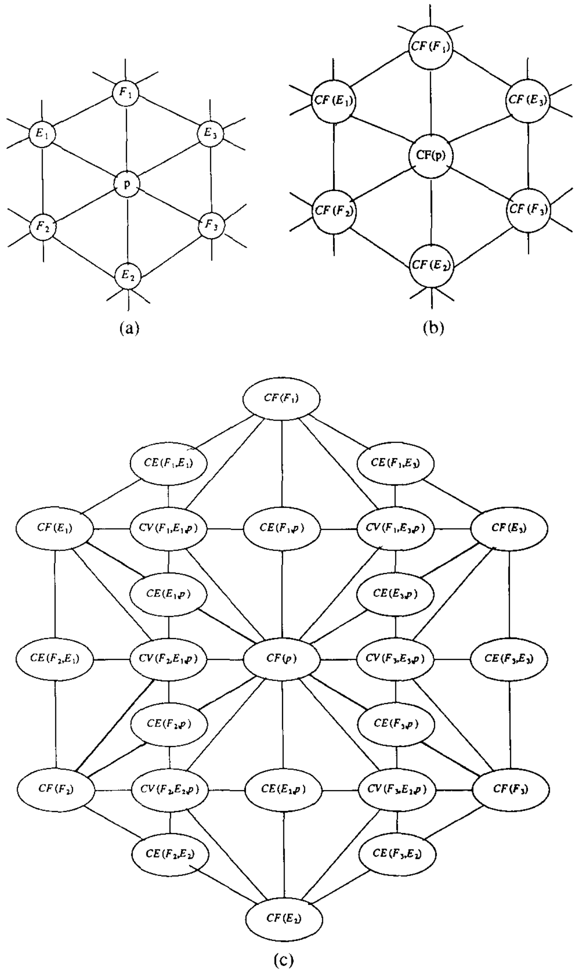

Fig. 3. (a) Adjacency graph for obstacle. (b) Weak topolgy for convolution faces. (c) Weak topology for convolution graph.

Edges of Convolution (S_r^2, p): When p is a convex, each boundary edge of conv (S_r^2, p) is the geodesic arc on the translated sphere $(S_r^2)_p$ connecting two points $p + r \cdot n_p$ and $p + r \cdot \hat{n}_p$, where n_p and $\hat{n}_p$ are unit outward normals of two adjacent trangular faces of the pyramid $\Lambda(p)$ at p, see Section IV-A for the definition of $\Lambda(p)$. Suppose a face F has a convex edge E and a concave edge E' both adjacent to a convex vertex p. Then, note that the convolution faces conv (S_r^2, E) and conv (S_r^2, p) do not share a common edge. An edge of conv (S_r^2, p) is properly contained in an edge of conv (S_r^2, E).

C. Generating Convolution Vertices

Each vertex $\bar{p}$ of the convolution face conv (S_r^2, K), where $K = F$ or a convex edge E, is a vertex of conv (S_r^2, F) for some face F, and $\bar{p} = p + r \cdot n_p$ for some vertex p of F, where n_p is the unit gradient of F at p. For a convex vertex p, each vertex $\bar{p}$ of the convolution face conv (S_r^2, p) is $\bar{p} = p + r \cdot n_p$ where n_p is the unit outward normal direction of a triangular face of the pyramid $\Lambda(p)$ at p.

D. Topology of Convolution $(S_r^2, Bdr(B))$

Having obtained all the faces, edges and vertices of conv $(S_r^2, Bdr(B))$, the next step is to connect these together with the correct topology. The topology of faces, edges, and vertices of B essentially induces a certain topology (weak topology) between the convolution faces conv (S_r^2, K), where $K = F, E$, or p. See Fig. 3 for the relationship between the adjacency graph for the obstacle and the weak topology for the convolution faces, edges, and vertices, where CF(K) means the convolution face generated by K, CE(K_1, K_2) means the convolution edge common to the convolution faces CF(K_1) and CF(K_2), and CV$(K_1, \cdots, K_k)$ means the convolution vertex common to the convolution faces CF$(K_1), \cdots,$ CF(K_k). However, this is not a complete adjacency graph. Some convolution faces which are not adjacent according to the weak topology may interfere because of the nonconconvexity of an obstacle B. Hence, to construct the convolution topology correctly, we also need to check all the intersections and self-intersections of convolution faces. This complete convolution topology will be represented in a graph, the convolution graph.

When two faces intersect or a face self-intersects, new edges and vertices are created for the intersection curve segments. The faces are subdivided into subfaces and these new faces are connected with appropriate adjacencies. Intersecting faces $f = 0$ and $g = 0$ can be detected by either numerically solving $f = g = 0$ or algebraically via resultants, Collins [12]. Self-intersections and singularities on surfaces can be computed algebraically by simultaneously solving $f = f_x = f_y = f_z = 0$ where f_x, f_y, and f_z are the x, y, and z partials. One can eliminate two variables from $f_x = f_y = f_z = 0$ [19], and check whether each solution satisfies $f = 0$. There is also the possibility that elimination of variables from $f_x = f_y = f_z = 0$ results in an expression which is identically 0. This occurs in the presence of a curve singularity on the surface. At this stage points on this curve singularity can be obtained by solving the equations numerically. Coincident faces are special cases of intersecting faces and are merged into a single face. Convolution faces which collapse into single edges or vertices are special cases of self-intersecting faces. The elimination of redundant faces, edges, and vertices is discussed in Section IV-E. While generating the convolution graph we tag each face, edge, and vertex resulting from the above singularities appropriately.

E. Removing Redundancies

Removing from the convolution graph the faces, edges, and vertices on which $A_{\bar{p}}$ and B intersect in an interior point ($A_{\bar{p}}$ collides with B), we obtain O-envelope (Theorem 2). Further removing various isolated edges and vertices, and coincident faces from O-envelope, one obtains the C-space obstacle boundary (Theorem 2). The process of obtaining the C-space obstacle boundary from the convolution graph is, however, more direct. The $Bdr(CO(A, B))$ we construct conforms to the boundary representation model of Section II. We note that even if B is an obstacle model without "holes," $Bdr(CO(A, B))$ may consist of more than one *peel*, corresponding to "holes" in the C-space obstacle.

In generating the C-space obstacle boundary, first the convolution graph is constructed as specified in Section IV-D. The intersecting and self-intersecting surface patches are broken up into faces with additions of new vertices and edges at the intersecting or singular curves. These are difficult to handle and at present only an approximate solution seems feasible. Next, a cleanup phase is initiated where redundancies such as isolated edges and vertices, coincident faces, and dangling faces are eliminated. These are either part of the conv $(S_r^2, Bdr(B))$ or the O-envelope, or are found while constructing the convolution graph. For example in the convolution graph construction phase, coincident faces of the O-envelope redundant to the C-space obstacle boundary may get merged into single dangling faces. Coincident faces are detected equationally by comparing coefficients, while isolated edges and vertices and dangling faces are detected from the condition that each edge of the convolution graph needs to be adjacent to exactly (nonzero) even number of faces. As a final cleanup step in the $Bdr(CO(A, B))$ generation one needs to eliminate the redundant faces, edges, and vertices which give rise to colliding configurations (Theorem 2).

From the way the convolution graph is constructed one can see that the relative configurations of $A_{\bar{p}}$ and B is constant on each face (i.e., either collide or contact for each point $\bar{p}$ of the face). Thus each convolution face can be classified as either a *collide* face or as a *contact* face by checking the relative configurations of $A_{\bar{p}}$ and B for a single point $\bar{p}$ in the interior of the face. To eliminate the redundant collide faces an edge-by-edge analysis then needs to be done. First consider an edge with only two adjacent faces. In this case, one can easily see that the adjacent face types in the convolution graph are either collide–collide or contact–contact. In the collide–collide case the common edge is redundant, and in the contact–contact case the common edge is in $Bdr(CO(A, B))$. Further, one notes that in a region of faces and edges on which each edge has only two adjacent faces, the whole region is either totally redundant or totally in $Bdr(CO(A, B))$. Thus one can classify the *simple* regions which have no edges with more than two adjacent faces except the boundary edges, as redundant or nonredundant. In summary, a search on the convolution graph with an edge-by-edge analysis as above allows one to delete all the redundant simple regions.

For edges with more than two adjacent faces the problem of deciding redundant regions is slightly more complicated. Such high valence edges arise either as a complex intersection of many surface patches or because of a self-intersection singularity. One needs a distinct point on each of the various faces incident to the edge for then a decision can be made as to whether the entire face is either redundant or nonredundant. By intersecting the adjacent faces and the common edge by a plane orthogonal to the edge in an interior point, we can reduce the distinct point generation problem on each face to a similar problem on each intersecting planar curve. Generating distinct points on various branches of an algebraic curve emanating from a singularity can be quite difficult for high order and irregular singularities, see Walker [28]. However, a local analysis about such high valence vertex points which yields distinct points on separate branches is always possible (Abhyankar [2]). Note that though this analysis applies to a singularity of a single algebraic curve with multiple branches at the singularity, for our purposes it can also be applied to the multiple edges arising from distinct intersecting curve segments. In this case one simply considers the product of all the distinct curve segments locally about the vertex point. By generating distinct points on the various edges incident on a high valence vertex and checking for collide configurations of $A_{\bar{p}}$ and B, the redundant faces can be detected and removed. Alternatively, a direct local analysis may be performed along the high valence edge which yields distinct points on the separate surfaces passing through the edge, Abhyankar [1]. However, this is a decisively more complicated procedure.

V. Simple Solids

In this section, we consider simple solids for which the convolution and the C-space obstacle generations are easy. Since many of the solids in practice fall into this category, the results in this section have practical importance. In Section V-A we consider the surfaces which preserve its type under the convolution operation. In Section V-B, we consider the convolution face generation for parametrizable edges. In Sections V-C, -D, we consider the C-space obstacle generations for the solids of revolution and the solids of linear extrusion. Convex solids of these types have also been considered by Farouki [14] although from a differential geometric viewpoint.

A. Simple Surfaces

1) When F is a patch of the plane $f(x, y, z) = ax + by + cz + d = 0$, conv (S_r^2, F) is a planar patch $\bar{f}(\bar{x}, \bar{y}, \bar{z}) = a\bar{x} + b\bar{y} + c\bar{z} + d - r(\sqrt{a^2+b^2+c^2}) = 0$.

2) When F is a patch of the cylindrical surface $f(x, y, z) = x^2 + y^2 - R^2 = 0$, conv (S_r^2, F) is a cylindrical patch $\bar{f}(\bar{x}, \bar{y}, \bar{z}) = (\bar{x})^2 + (\bar{y})^2 - (R + r)^2 = 0$.

3) When F is a patch of the spherical surface $f(x, y, z) = x^2 + y^2 + z^2 - R^2 = 0$, conv (S_r^2, F) is a spherical patch $\bar{f}(\bar{x}, \bar{y}, \bar{z}) = (\bar{x})^2 + (\bar{y})^2 + (\bar{z})^2 - (R + r)^2 = 0$.

4) When F is a patch of the conic surface $f(x, y, z) = x^2 + y^2 - z^2 = 0$ without the apex on it, conv (S_r^2, F) is a patch of the conic surface $\bar{f}(\bar{x}, \bar{y}, \bar{z}) = (\bar{x})^2 + (\bar{y})^2 - (\bar{z} + \sqrt{2}\, r)^2 = 0$.

5) When F is a patch of the toridal surface $f(x, y, z) = (\sqrt{x^2 + y^2} - R_1)^2 + z^2 - R_2 = 0$ with $R_1 > R_2 + r$, conv (S_r^2, F) is a patch of the toroidal surface $\bar{f}(\bar{x}, \bar{y}, \bar{z}) = (\sqrt{\bar{x}^2 + \bar{y}^2} - R_1)^2 + \bar{z}^2 - (R_2 + \bar{r})^2 = 0$.

6) In general, when F is a Dupin cyclide surface patch, conv (S_r^2, F) is a Dupin cylcide surface patch, see Martin [20].

B. Analytically Parametrizable Edges

We describe an analytic method to generate the convolution face for any parametrizable edge where the parametrization is given by some analytic function in one variable. Suppose the common edge E of faces F and $\hat{F}$ is parametrizable by $\gamma(t)$ for $a \leq t \leq b$, the inner angles between F and $\hat{F}$ along E are > 0 and $< \pi$, and the gradients of F and $\hat{F}$ along E are given by $n(t)$ and $\hat{n}(t)$ for $a \leq t \leq b$, then the convolution face conv (S_r^2, E) is given by some analytic function of $n(t)$ and $\hat{n}(\hat{t})$ as follows. Let $h_t(s) = (1 - s)\cdot n(t) + s\cdot\hat{n}(t)$ for $0 \leq s \leq 1$, then $h_t(s) \neq 0$ and conv (S_r^2, E) is given by the parametric surface $H(s, t) = \gamma(t) + r\cdot h_t(s)/\|h_t(s)\|$ for $0 \leq s \leq 1$ and $a \leq t \leq b$.

C. Solids of Revolution

A solid of revolution is obtained by rotating a planar area about the axis of revolution. We may assume this planar area is bounded by a Jordan curve and totally contained in a closed half-plane bounded by the revolution axis. One can easily show the C-space obstacle is also a solid of revolution and the generating planar area is the intersection of the half plane and the planar C-space obstacle generated by moving a circle of radius r around the original planar generating area. For the planar C-space obstacle generation one can use the results from Bajaj and Kim [8], where the C-space obstacle generation for nonconvex planar moving object and obstacles with algebraic curve boundaries is discussed.

D. Solid of Linear Extrusion

A solid of linear extrusion is obtained by sweeping a planar area from the bottom face to the top face along the normal direction. We may assume this planar area is bounded by a Jordan curve. The boundary surface generated by sweeping this boundary Jordan curve is called the side walls. The convolution faces generated by the top and bottom faces are obtained by simply translating these faces along the normal direction by a distance r. One can easily show that the convolution faces generated by the side walls are obtained by sweeping a Jordan curve which is the planar C-space obstacle boundary obtained from the original sweeping planar area and the circle of radius r. There are also convolution faces generated by the top (resp. bottom) edges between the top (resp. bottom) faces and the side walls, and the convolution faces generated by the vertices between these edges. The convolution faces generated by top (resp. bottom) edges have one of its boundary edge on the boundary of the top (resp. bottom) face offset and one on the boundary of the side walls convolution. The top and bottom face offsets have no redundancies, but the side walls convolution may have redundancies when the generating planar area is nonconvex. Hence the convolution faces generated by top and bottom edges cannot be totally redundant, but some subregions of these faces may be redundant. These partial redundancies are detected by the redundancies of the corresponding side walls offset. The convolution faces generated by vertices are either nonredundant or totally redundant depending on whether the corresponding vertex on the generating planar area is either convex or concave. By removing these redundancies and computing the common edges between partially redundant convolution faces one can construct the C-space obstacle boundary correctly.

VI. Relations to Blending

Nonsmoothness on the C-space obstacle boundary results from the intersections and self-intersections of the convolution faces. The connected subregions of the C-space obstacle boundary where no intersections or self-intersections of the convolution faces lie are smooth surface patches. Intersections and self-intersections mean there are at least two contact points for a moving sphere when its center is placed at these intersections, and these also mean there are global or local concave regions on the obstacle which prohibit a moving sphere of radius r to roll over some region. When a sphere can roll over all the points of a region Γ on the obstacle while making contacts with no more than one point, conv (S_r^2, Γ) itself is a smooth surface patch. See Rossignac and Requicha [23] and Hoffman and Hopcroft [16] where this is used to derive surface patches smoothing out a surface intersection edge or a vertex.

VII. Conclusion

We have described algorithmic methods to generate the boundary of configuration space obstacles arising from the motion of a sphere among obstacles. The boundaries of the obstacles are given by patches of algebraic surfaces. Algorithms are given for both implicit and parametric surface patches. Both convex and nonconvex obstacles are considered. In the case of convex obstacles, the topology of convolution faces is the same as the adjacency graph of faces, edges, and vertices of the obstacle. Further, there are no redundancies in the convolution faces. Redundancies on the convolution can occur in the case of nonconvex obstacles. One may detect these redundancies from the intersections and self-intersections of convolution faces. We also consider simple solids for which the convolution and the C-space obstacle generation is easy.

One possible extension of this research is for the case of translatory motion of arbitrary nonconvex object among nonconvex obstacles. In this general case, each face, edge, and vertex of an obstacle can interact with many different faces, edges, and vertices of the moving object and can generate many convolution faces. The interconnection of the convolution faces becomes more complicated. Further, the definition of outward normal directions would have to be modified. This differs from the case of a moving sphere where a sphere cannot touch concave edges or concave vertices.

References

[1] S. Abhyankar, "Weighted expansions for canonical desingularization," in *Lecture Notes in Mathematics,* no. 910. New York: Springer-Verlag. 1982.

[2] S. Abhyankar, "Desingularization of plane curves," in *Proc. Symp. Pure Mathematics,* vol. 40, pp. 1-45, 1983.

[3] S. Abhyankar and C. Bajaj, "Automatic rational parameterization of curves and surfaces II: Cubics and conicoids," *Computer Aided Design,* vol. 19, pp. 11-14, 1987.

[4] ——, "Automatic rational parameterization of curve and surfaces II: Cubics and cubicoids," *Computer Aided Design,* vol. 19, pp. 499-502, 1987.

[5] ——, "Automatic ration parameterization of curves and surfaces III: Algebraic plane curves." Purdue Univ., W. Lafayette. IN, Comput. Sci. Tech. Rep. CSD-TR-619, 1987.

[6] C. Bajaj, "Algorithmic inplicitization of rational curves and surfaces," Purdue Univ., W. Lafayette, IN, Comput. Sci. Tech. Rep. CSD-TR-681.

[7] C. Bajaj and M. S. Kim, "Generation of configuration space obstacles II: The case of moving algebraic surfaces," Purdue Univ., W. Lafayette, IN, Comput. Sci. Tech. Rep. CSD-TR-586, 1986.

[8] ——, "Generation of configuration space obstacles: The case of moving algebraic curves," in *Proc. 1987 IEEE Int. Conf. Robotics and Automation*, 1987, pp. 979-984 (updated version to appear in *Algorithmica).*

[9] ——, "Compliant motion planning with geometric models," in *Proc. 3rd ACM Symp. Computational Geometry,* Waterloo, Canada, 1987, pp. 171-180.

[10] J. Canny, "The complexity of robot motion planning," Ph.D. dissertation, Dep. Elec. Eng. Comput. Sci., Mass. Inst. Technol., Cambridge, 1987.

[11] B. Chazelle, "Convex partitions of polyhedra: A lower bound and worst-case optimal algorithm," *SIAM J. Computing,* vol. 13, pp. 488-507, 1984.

[12] G. Collins, "The calculation of multivariate polynomial resultants," *J. Assoc. Comput. Mach.,* vol. 18, pp. 515-532, 1971.

[13] B. Donald, "Motion planning with six degress of freedom," Mass. Inst. Technol., AI Tech. Rep. 791, 1984.

[14] R. Farouki, "Exact offset procedures for simple solids," *Computer Aided Geometric Design,* vol. 2, pp. 257-279, 1985.

[15] ——, "The approximation of non-degenerate offset surfaces," *Computer Aided Geometric Design,* vol. 3, pp. 15-43, 1986.

[16] C. Hoffman and J. Hopcroft, "The potential method for blending surfaces and corners," Comput. Sci., Cornell Univ., Ithaca, NY, TR85-699, 1985.

[17] T. Lozano-Perez, "Spatial planning: A configuration space approach," *IEEE Trans. Comput.,* vol. C-32, pp. 108-120, 1983.

[18] T. Lozano-Perez and M. Wesley," An algorithm for planning collision free paths among polyhedral obstacles," *Commun. Assoc. Comput. Mach.,* vol. 22, pp. 560-570, 1979.

[19] F. Macaulay, "Some formulae in elimination," *Proc. London Math. Soc.,* vol. 1, pp. 2-27, 1903.

[20] R. Martin, "Principal patches—A new class of surface patch based on differential geometry," in *Proc. Eurographics '83,* P. J. W. ten Hagen, Ed. Amsterdam, The Netherlands: North-Holland, 1983.

[21] A. Requicha, "Representations of rigid solid objects," in *Springer Lecture Notes in Computer Science,* no. 89. New York: Springer-Verlag, 1980, pp. 2-78.

[22] A. Requicha and H. Voelcker, "Solid modeling: Current status and research directions," *IEEE Comput. Graphics Appl.,* pp. 25-37, 1983.

[23] J. Rossignac and A. Requicha, "Constant-radius blending in solid modeling," *Comput. in Math Eng.,* pp. 65-73, 1984.

[24] ——, "Offsetting operations in solid modeling," *Computer Aided Geometric Design,* vol. 3, pp. 129-148, 1986.

[25] W. Tiller and E. Hanson, "Offsets of two-dimensional profile," *IEEE Comput Graphics Appl.,* pp. 36-46, Sept. 1984.

[26] S. Udupa, "Collision detection and avoidance in computer controlled manipulations," in *Proc. 5th Int. Joint Conf. Artificial Intelligence,* Cambridge, MA, 1977, pp. 737-748.

[27] B. van der Waerden, *Modern Algebra.* New York: Ungar, 1950.

[28] R. Walker, *Algebraic Curves.* New York: Springer-Verlag, 1978.

The Dexterous Workspace of Simple Manipulators

ZONE-CHANG LAI AND CHIA-HSIANG MENQ

Abstract—**A theoretical study on the dexterous workspace of robotic manipulators is presented. For a robot with wrists which can generate a full range of orientations, the boundary of the robot's dexterous workspace is governed by the boundary of $W_1(4)$, where $W_1(4)$ is the reachable space of joint 4 when joints 1-3 are free to rotate. Based on this concept, a method was developed. Three examples are given to illustrate this concept and method. For simple robots, as demonstrated by three examples, analytical expressions of the dexterous workspace may be obtained using the method presented.**

I. Introduction

The determination of extreme positions of the end-effector of a manipulator and the evaluation of workspace have been the subject of many investigations [1]-[13]. However, the determination of dexterous workspace, due to the complicated relationship between hand position and orientation, has been investigated for those robots with wrists having last three joint axes coincide. Kumar and Waldron [4] introduced the concept of dexterous workspace (a space in which the manipulator's hand can rotate fully about all axes through any point)

Manuscript received May 2, 1985; revised July 28, 1986.

Z. C. Lai was with GMF Robotics Corporation, Troy, MI 48098. He is now with Allen-Braley Co., Highland Heights, OH 44143.

C. H. Menq is with the Department of Mechanical Engineering, Ohio State University, Columbus, OH 43210.

IEEE Log Number 8718079.