Compliant Motion Planning with Geometric Models†

Chanderjit Bajaj and Myung-Soo Kim

Department of Computer Science,
Purdue University,
West Lafayette, IN 47907.

Abstract: We present algebraic algorithms to generate the boundary of configuration space obstacles arising from the translatory motion of objects amongst obstacles. In particular we consider obtaining compliant motion paths where a curved convex object with fixed orientation moves in continuous contact with the boundary of curved convex obstacles in three Dimensions. Both the boundaries of the objects and obstacles are given by patches of algebraic surfaces. We also give a method to obtain approximate geodesic paths on convex C−space obstacles with algebraic boundary surfaces.

1. Introduction

Using configuration space, (*C−Space*), to plan motion for a single rigid object amongst physical obstacles, reduces the problem to planning motion for a mathematical point amongst "grown" configuration space obstacles, (the points in *C−Space* which correspond to the object overlapping one or more obstacles), Lozano-Perez (1983). For example, a rigid polyhedral object in compliant motion, viz., in continuous contact with the boundary of obstacles in 3-Dimensions can be represented as a point constrained to move on the three (or higher) dimension boundaries of grown obstacles embedded in 6-Dimension *C−Space*, Donald (1984). Compliant motion is then simply moving a point on connected boundary regions of the configuration space obstacles, (*C−space* obstacles), Lozano-Perez, Mason and Taylor (1984), Sharir and Schorr (1984), Franklin and Akman (1984), O'Rourke, Suri and Booth (1985), Mount (1985), Mitchell, Mount and Papadimitriou (1985). The technique thus relies, (and this is in general the more difficult part), in efficiently generating the boundary of *C−Space obstacles*. Numerous applications such as fine motion strategies for assembly, machining parts, etc., exist where compliant motion proves necessary, Lozano-Perez, Mason and Taylor (1984).

Early uses of the configuration space approach were, Freeman (1975), Udupa (1977), and more recently, Lozano-Perez and Wesley (1979), Lozano-Perez (1983), Lozano-Perez, Mason and Taylor (1984), Schwartz and Sharir (1983), Sharir and Schorr (1984), Franklin and Akman (1984), Canny (1984), Donald (1984), O'Rourke, Suri and Booth (1985), Mount (1985), Mitchell, Mount and Papadimitriou (1985), Yap (1985), Bajaj and Kim (1986, 87). The only efficient algorithms known for generating *C−Space* obstacles have been for polyhedral (degree 1) surface objects and obstacles, using methods for efficiently computing convex hulls, Lozano-Perez (1983), and recently efficient convolution algorithms for Minkowski addition, Guibas and Seidel (1986). However it has progressively become easier for geometric modeling systems to deal with objects that are defined by quadrics (degree 2) and higher degree surfaces, Requicha and Voelcker (1983). Further, motion planning in these sophisticated modeling environments for example for process simulation, Hopcroft and Krafft (1985), suggests the need to characterize and efficiently generate the surface boundary of *C−Space obstacles* arising from the motion of objects amongst obstacles with curved surface boundaries.

The methods based on generating a cylindrical cell decomposition of free *C−Space*, though applicable for general objects and obstacles defined by semi-algebraic sets, are computationally too restrictive, Schwartz and Sharir (1983), Yap (1985). Thus, in the past the object representations that have been considered for planning motion in three dimensions, have been polyhedral approximations to the curved object. However if we approximate object and obstacle by polyhedra, then for example, the compliant motion paths obtained are "jumpy" or provide discrete contact motion. One solution to obtaining a continuous compliant motion includes generation of the curved surface boundary of the *C−space* obstacle. As we show in this paper, one needs to generate only the specific part of the *C−space* obstacle boundary which contains the desired compliant path. Algebraic approximate shortest paths (approximate geodesics) can then also be obtained by projecting on the curved *C−space* obstacle the shortest path obtained on approximated polyhedral *C−space* obstacles.

The main contributions of this paper are as follows. In §3 we show that the boundary of *C−Space obstacles* for general curved objects moving with only translation can be viewed as either the convolution between the obstacle boundary and the reversed object boundary (reversed with respect to a reference point on the object) or by computing certain *envelopes* of boundary surfaces of the moving reversed object with the reference point moving on the physical obstacle. Next in §4 we give algebraic algorithms to generate the curves and surfaces which make up the boundary of the three dimensional *C−Space obstacles*. Here we only consider objects and obstacles which are convex. These objects and obstacles are represented by a general algebraic boundary representation model discussed in §2. Crucial too here is the internal representation of curves and surfaces, i.e., whether they are parametrically or implicitly defined†. We present algorithms for both these internal representations. Further in §5 we show how to construct the topology of the *C−space* obstacle boundary. Use is made of a Gaussian (spherical) model discussed in §2.

† Research supported in part by NSF grant DCI-85 21356 and a David Ross Fellowship.

† A unit circle is implicitly given as $x^2+y^2-1=0$ and in rational parametric form as $x=(1-t^2)/(1+t^2)$ and $y=2t/(1+t^2)$.

In §6 we consider motion of a point on the boundary of *C-space* obstacles. With only translational parameters for the moving object one essentially considers compliant motion wherein the contact points between object and obstacle change during motion. Requiring the contact points to remain the same during compliant motion necessitates the introduction of rotational parameters and thus higher dimension *C-spaces*. For compliant motion we give algebraic algorithms to compute paths on curved surfaces in three dimensions. Locally shortest or geodesic curves on algebraic surfaces are also considered. Exact algebraic algorithms for geodesics are impossible in general because of the existence of non-algebraic or transcendental geodesic curves. Here we also introduce a Gaussian polyhedral approximation model which allows efficient algebraic approximations of geodesic paths on curved surfaces.

2. Geometric Models

2.1. Solid Algebraic Model

In a boundary representation an object with general algebraic surfaces 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 (1980). Geometric disambiguation may also be achieved by the methods of Hoffmann and Hopcroft (1986).

(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 non-singular except at vertices and edges, two distinct faces intersect only in edges, etc. The objects and obstacles that we consider are *solids* and are assumed to enclose non-zero finite volume. Hence non-regularities such as dangling edges and dangling faces which depending on one's viewpoint enclose zero or infinite volume, are not permitted. The *C-spaces* that we construct are also regularized in this fashion and assumed to be solids enclosing non-zero finite volume.

2.2. Gaussian Model

Let S^2 be the unit sphere in R^3, and $Bdr(S)$ be the boundary surface of a convex set $S \subset R^3$. $Bdr(S)$ is homeomorphic to S^2. The *Gaussian Map* of S is defined as follows. For any set $K \subset Bdr(S)$, we shall define a set $N(S,K) \subset S^2$ as follows. A point $e \in S^2$ belongs to $N(S,K)$ if there exists a point $p \in K$ and a supporting plane L_p at p such that the exterior normal to L_p translated to the center of S^2 has e as its end point. This set $N(S,K)$ is called the *Gaussian Image* of K. The function $N(S,\cdot) : P(Bdr(S)) \to P(S^2)$ is called the *Gaussian Map* of S, where $P(Bdr(S))$ and $P(S^2)$ are the power sets of $Bdr(S)$ and S^2. It is a bijective map and its inverse $N^{-1}(S,\cdot) : P(S^2) \to P(Bdr(S))$ is called the *Inverse Gaussian Map* of S. For any $G \subset S^2$, the *Inverse Gaussian Image* of G is defined as $N^{-1}(S,G)$. The *Gaussian Curvature* of $p \in Bdr(S)$ is the limit of the ratio (Area of $N(S,K)$) / (Area of K) as K shrinks to the point p, see Pogorelov (1978), Horn (1986).

Gaussian Image of Faces, Edges and Vertices

Since all faces are patches of algebraic surfaces, we may assume that each face is either a strictly convex face (Gaussian curvature is positive on each point), a ruled surface patch, or a planar patch. The Gaussian Model of a curved object then consists of a finite set of vertices, edges and faces on the surface of a unit sphere as follows.

(1) For a strictly convex face F, the Gaussian Image $N(S,F)$ is a patch of S^2 with its boundary curves determined by the normals to the tangent planes of F at the boundary. That is, the boundary of $N(S,F)$ consists of the set of points $\nabla f(p) / ||\nabla f(p)||$ for $p \in \bigcup_{E \in \Gamma} E$, where Γ is the set of boundary edges of F. For a ruled surface patch F, $N(S,F)$ is a degenerate curve on S^2. And for a planar patch F, $N(S,F)$ is a degenerate point on S^2.

(2) For an edge E, there are two faces F and G intersecting in E. By subdividing E if necessary, we may assume that F and G either cross each other along E or are tangent to each other along E. When F and G cross each other, each point $p \in E$ determines two different points n_F and n_G on S^2 determined by the exterior normals of the tangent planes of F and G at p. $N(S,p)$ is the geodesic arc γ_p connecting n_F and n_G on S^2 and $N(S,E) = \bigcup_{p \in E} \gamma_p$ is a patch of S^2. $N(S,E)$ has 4 boundary curves, one is the set of points $\nabla f(p) / ||\nabla f(p)||$ for $p \in E$, one is the set of points $\nabla g(p) / ||\nabla g(p)||$ for $p \in E$, and the others are the geodesic arcs γ_{p_1} and γ_{p_2}, where $f = 0$ and $g = 0$ are the surface equations of F and G, and p_1 and p_2 are two end points of E. When F and G are tangent to each other along E, $N(S,E)$ is a degenerate curve on S^2. $N(S,E)$ is the common boundary curve of $N(S,F)$ and $N(S,G)$. That is, it is the set of points $\nabla f(p) / ||\nabla f(p)|| = \nabla g(p) / ||\nabla g(p)||$ for $p \in E$, When F and G are planar patches, E is a linear edge and $N(S,E)$ is a degenerate geodesic arc γ connecting n_F and n_G on S^2, where n_F and n_G are the exterior normals of F and G.

(3) For a vertex p, suppose that there are k adjacent faces (ordered in a counter-clockwise direction) $F_1, F_2, ..., F_k$ intersecting at p. Each face F_i determines a point n_i on S^2 determined by the normal of F_i at p. Let γ_i $(i = 1, ..., k)$ be the geodesic arc (greatest circle) on S^2 connecting n_i and n_{i+1} where $n_{k+1} = n_1$. Then $N(S,p)$ is the convex patch on S^2 bounded by the cycle of geodesic arcs $\gamma_1, \gamma_2, ..., \gamma_k$. When F_i and F_{i+1} is tangent on p, γ_i is a degenerate point. In the special case of all k faces being tangent at p, $N(S,p)$ is a degenerate point. $N(S,p)$ can also be a degenerate geodesic arc on S^2 when $Bdr(S)$ is locally singular only along a curve which is tangent at p. Otherwise, $N(S,p)$ is a patch on S^2.

Topology of Gaussian Model

The Gaussian Image of $Bdr(S)$ covers S^2 completely and subdivides S^2 into faces, edges and vertices as described above. We shall fudge the physical distinctions of face, edge and vertex of S^2 a little bit and deal with the degenerate edges and vertices in the same way as with the faces. Let us assume the Gaussian Image of each face, edge and vertex is a generic face of S^2. If any of these Gaussian Images are not faces, we can represent this fact by tagging it as degenerate curves or degenerate points and consider it as faces. By using the connectivity graph of $Bdr(S)$ we can connect these generic faces with the correct topology. We can further include the edges and vertices determined by these faces into the connectivity graph of the Gaussian Image. The edge equations and vertex coordinates are given by the face boundary equations described above. Doing it in this way, we construct a graph on S^2 with degenerate curves and points considered as generic faces tagged appropriately.

Figure 1 (b) and (d) show the Gaussian Models for the convex objects in Figure 1 (a) and (c). In Figure 1 (a), all the faces are strictly convex, and all the edges and vertices are singular edges and singular vertices. Hence, all the corresponding Gaussian Images are patches of S^2. In Figure 1 (c), the face F_3 is a ruled surface and the face F_2 is a planar patch. The corresponding Gaussian Images are a degenerate curve and a degenerate point. Further since faces F_1 and F_3 are tangent to each other along E_2, the Gaussian Image of E_2 is a degenerate curve.

3. C-space Obstacles, Convolution and Envelopes

Let A be a moving object with its reference point at the origin and B be a fixed obstacle in the 3-dimensional real Euclidean plane R^3. Both A and B are modeled by the above boundary representations. For the sake of notation and preciseness in our usage we make the following distinctions. We denote $Int(A)$ as the interior of A and $Bdr(A)$ as the boundary of A. Note that $A = Int(A) \cup Bdr(A) = Cl(A)$ = closure of A by regularity. Further, the exterior of A is denoted by $Ext(A) = A^c$ (the complement of A) $= R^3 - A$, where the set difference $P - Q = \{p \in R^3 \mid p \in P$ and $p \notin Q\}$. Note that $Int(A)$ and $Ext(A)$ are open sets.

Throughout we consider object A to be free to move with fixed orientation. In this case configuration space is also 3-dimensional. We fix a reference point on A and denote A_p to be A located in R^3 with its reference point at the point $p \in R^3$. We also have $d(p,q)$ as the Euclidean distance between p and q; $NB_\varepsilon(p) = \{q \in R^3 \mid d(p,q) < \varepsilon\}$ = ε-neighborhood around a point p; $-A = \{-p \mid p \in A\}$ = Minkowski inverse, $A \pm B = \{p \pm q \mid p \in A$ and $q \in B\}$ = Minkowski sum and difference.

One also needs the following distinctions (1) $A_{\bar{p}}$ is *free* from B $\Longleftrightarrow$ $A_{\bar{p}} \cap B = \emptyset$. (2) $A_{\bar{p}}$ *collides* with B $\Longleftrightarrow$ $Int(A_{\bar{p}}) \cap Int(B) \neq \emptyset$ (3) $A_{\bar{p}}$ *contacts* with B $\Longleftrightarrow$ $A_{\bar{p}} \cap B \neq \emptyset$ and $Int(A_{\bar{p}}) \cap Int(B) = \emptyset$ (Note that these conditions imply $Bdr(A_{\bar{p}}) \cap Bdr(B) \neq \emptyset$.) (4) $CO(A,B)$ = *C-space* obstacle due to A and B = $\{\bar{p} \in R^3 \mid A_{\bar{p}} \cap B \neq \emptyset\}$. (5) *O-Envelope*$(-A,B)$ = Outer envelope due to $-A$ and B = $\{\bar{p} \in R^3 \mid \bar{p} \in Bdr((-A)_p)$ for some $p \in Bdr(B)$, and $\bar{p} \notin Int((-A)_q)$ for any $q \in B\}$ (Having $q \in B$ as opposed to $q \in Bdr(B)$ implies that only the outer envelope is considered.) (6) *Convolution*$(Bdr(-A), Bdr(B))$ = Convolution of $Bdr(-A)$ and $Bdr(B)$ = $\{\bar{p} \in R^3 \mid \bar{p} = p - q$ where $p \in Bdr(B)$ and $q \in Bdr(A)$ and B has an outward normal direction at p exactly opposite to an outward normal A has at $q\}$.

We now note the following.

Theorem 3.1 : $CO(A,B) = B - A$

Proof: Lozano-Perez and Wesley (1979). □

From the above Theorem and our prior definitions we obtain,

Corollary 3.2 : (1) $CO(Int(A), Int(B)) = Int(B) - Int(A) = B - Int(A)$ (This is an open set)

(2) $A_{\bar{p}}$ is *free* from B $\Longleftrightarrow$ $\bar{p} \in Ext(CO(Int(A), Int(B)))$

(3) $A_{\bar{p}}$ *collides* with B $\Longleftrightarrow$ $\bar{p} \in Int(CO(Int(A), Int(B)))$

(4) $A_{\bar{p}}$ *contacts* with B $\Longleftrightarrow$ $\bar{p} \in Bdr(CO(Int(A), Int(B)))$

We next obtain the following important characterizations,

Lemma 3.3 : $Bdr(CO(Int(A), Int(B))) = O\text{-}Envelope(-A,B)$

Proof : ($\subseteq$) : Let $\bar{p} \in Bdr(CO(Int(A), Int(B)))$, then $A_{\bar{p}}$ *contacts* with B, (*Corallary* 3.2 (4)), and $\exists\, p \in Bdr(A_{\bar{p}}) \cap Bdr(B)$. Since $p - \bar{p} \in Bdr(A)$, we have $\bar{p} - p \in Bdr(-A)$ and $\bar{p} \in Bdr((-A)_p)$ for $p \in Bdr(B)$. Further $\bar{p} \notin Int((-A)_q)$ for any $q \in B$. Assuming the contrary, if $\bar{p} \in Int((-A)_q)$ for some $q \in B$, then $\bar{p} \in B - Int(A) = Int(B) - Int(A) = Int(CO(Int(A), Int(B)))$, (contradiction).

($\supseteq$) : Let $\bar{p} \in O\text{-}Envelope(-A,B)$, then $\bar{p} \in Bdr((-A)_p)$ for some $p \in Bdr(B)$, and $\bar{p} \notin Int((-A)_q)$ for any $q \in B$. Equivalently, $p \in Bdr(A_{\bar{p}}) \cap Bdr(B)$ and $q \notin Int(A_{\bar{p}})$ for any $q \in B$. This implies $A_{\bar{p}} \cap B \neq \emptyset$ and $Int(A_{\bar{p}}) \cap Int(B) = \emptyset$. Hence, $A_{\bar{p}}$ contacts with B. □

Theorem 3.4 : $Bdr(CO(A,B)) \subset O\text{-}Envelope(-A,B) \subset Convolution(Bdr(-A), Bdr(B))$

Proof : (1) Using Theorem 3.3 we show $Bdr(CO(A,B)) \subset Bdr(CO(Int(A), Int(B)))$: For any $\bar{p} \in CO(A,B)$, $A_{\bar{p}} \cap B \neq \emptyset$, equivalently $\bar{p} \in Cl(CO(Int(A), Int(B)))$, (*Corollary* 3.2 (2)). Hence, $CO(Int(A), Int(B)) \subset CO(A,B) \subset Cl(CO(Int(A), Int(B)))$ and $Cl(CO(A,B)) = Cl(CO(Int(A), Int(B)))$. Since $Int(CO(Int(A), Int(B))) \subset Int(CO(A,B))$, we have $Bdr(CO(A,B)) \subset Bdr(CO(Int(A), Int(B)))$.

(2) $O\text{-}Envelope(-A,B) \subset Convolution(Bdr(-A), Bdr(B))$: For any $\bar{p} \in O\text{-}Envelope(-A,B) = Bdr(CO(Int(A), Int(B)))$, since $A_{\bar{p}}$ contacts with B at some $p \in Bdr(B)$, $A_{\bar{p}}$ has an outward normal direction at p which is opposite to an outward normal direction B has at p. For $q = p - \bar{p} \in Bdr(A)$, we have $\bar{p} = p - q$ and B has an outward normal direction at p exactly opposite to an outward normal A has at q. Thus $\bar{p} \in Convolution(Bdr(-A), Bdr(B))$. Also see Guibas, Ramshaw, and Stolfi (1983). □

In the special case when both A and B are convex, both the set containments of Theorem 3.4 become equalities. This follows from the properties of convexity. In particular we use the following simple fact. For convex A and B, if $A_{\bar{p}}$ and B have opposite outward normal directions at $p \in Bdr(A_{\bar{p}}) \cap Bdr(B)$, then there is a common supporting plane P_p such that $A_{\bar{p}}$ and B are on opposite sides of the plane P_p, Kelly and Weiss (1979).

Theorem 3.5 : For convex A and B, we have $Bdr(CO(A,B)) = O\text{-}Envelope(-A,B) = Convolution(Bdr(-A), Bdr(B))$ and is convex.

Proof : Using Theorem 3.4, all we need to show is $Convolution(Bdr(-A), Bdr(B)) \subset Bdr(CO(A,B))$ for convex A and B. Suppose $\bar{p} \in Convolution(Bdr(-A), Bdr(B))$. We first show $\bar{p} \notin Ext(CO(A,B))$. If $\bar{p} \in Ext(CO(A,B))$, then $\exists\, \varepsilon > 0$ such that $(A_{\bar{p}} + NB_\varepsilon(0)) \cap B = \emptyset$ and $Cl(A_{\bar{p}}) \cap Cl(B) = \emptyset$. Hence, $\bar{p} \notin Bdr((-A)_p)$ for any $p \in Bdr(B)$, (contradiction), and so $\bar{p} \notin Ext(CO(A,B))$. Now, we show $\bar{p} \notin Int(CO(A,B))$. Since $\exists\, p \in Bdr(A_{\bar{p}}) \cap Bdr(B)$ such that $A_{\bar{p}}$ and B have opposite outward normal directions at p, a common supporting plane P_p separates $A_{\bar{p}}$ and B. For any $\varepsilon > 0$, let e be an outward normal vector to B at p such that $\| e \| = \varepsilon$ and e is orthogonal to P_p, then $A_{(\bar{p}+e)}$ and B are separated by the banded region bounded by $P_{(\bar{p}+e)}$ and P_p, and so $A_{(\bar{p}+e)} \cap B = \emptyset$. Hence, $\bar{p} \notin Int(CO(A,B))$. Thus $\bar{p} \notin Int(CO(A,B)) \cup Ext(CO(A,B))$ implies $\bar{p} \in Bdr(CO(A,B))$. □

For convex A and B, the faces of $Convolution(Bdr(-A), Bdr(B))$ have no singularities in the interior with possibility of singularities occuring only at the vertices and edges of *C-space* obstacle boundary. Singular vertices can occur only from singular vertex–vertex contacts between A and B. Singular edges can occur either from singular vertex–edge contacts or *parallel* singular edge–edge contacts between A and B. This may then suggest a natural method for handling *non-convex* object and obstacle shapes. One first obtains a convex decomposition consisting of the union of convex pieces and then generates the *C-space* obstacle as the union of *C-space* obstacles for convex object and obstacle pairs. Such convex decompositions are possible for polyhedral objects, see Chazelle (1984). However not all objects with algebraic curve and surface boundaries permit decompositions consisting of the union of convex pieces – for example a torus. 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 (1983). Thus for the time being one is restricted to considering convex shaped objects and obstacles.

4. Generating the Boundary of *C-space* Obstacles

Suppose S be $-A$ or B, $p \in Bdr(S)$ be a boundary point, $E \subset Bdr(S)$ be an edge, and $F \subset Bdr(S)$ be a face. Let (F_S, N_{F_S}) be a pair such that $F_S \subset Bdr(S)$ is a face and $N_{F_S} = N(S, F_S)$, where $N(S, \cdot)$ is the *Gaussian Map* of S. (E_S, N_{E_S}) be a pair such that $E_S \subset Bdr(S)$ is an edge and $N_{E_S} \subset N(S, E_S)$ with $N_{E_S} \cap N(S,p) \neq \emptyset$ for all $p \in E_S$. (p_S, N_{p_S}) be a pair such that $p_S \in Bdr(S)$ is a vertex and $N_{p_S} \subset N(S, p_S)$ with $N_{p_S} \neq \emptyset$. Further let K_B be F_B, E_B or p_B, and let G_{-A} be F_{-A}, E_{-A} or p_{-A}. There are nine (K_B, G_{-A}) pairs. We define *sub-compatible* and *compatible* pairs as follows.

(1) K_B and G_{-A} are *sub-compatible* $\iff N(B, K_B) \cap N(-A, G_{-A}) \neq \emptyset$

(2) (K_B, N_{K_B}) and $(G_{-A}, N_{G_{-A}})$ are *compatible* $\iff N_{K_B} = N_{G_{-A}}$

Further denote by $K_B \infty G_{-A}$ that K_B and G_{-A} are *sub-compatible*. Since only sub-compatible pairs can contribute to the *Convolution*, one can show that $Convolution(Bdr(-A), Bdr(B)) = \bigcup_{K_B \infty G_{-A}} Convolution(G_{-A}, K_B)$. We can further refine the right-hand side to be a union of only the *compatible* pairs as follows. For a sub-compatible (K_B, G_{-A}) pair, let $N(K_B, G_{-A}) = N(B, K_B) \cap N(-A, G_{-A})$ be the nonempty intersection of two Gaussian Images of K_B and G_{-A}. $K(K_B, G_{-A}) = N^{-1}(B, N(K_B, G_{-A})) \subset K_B$ and $G(K_B, G_{-A}) = N^{-1}(-A, N(K_B, G_{-A})) \subset G_{-A}$ be the Inverse Gaussian Images of $N(K_B, G_{-A})$. Then $(K(K_B, G_{-A}), N(K_B, G_{-A}))$ and $(G(K_B, G_{-A}), N(K_B, G_{-A}))$ are *compatible*. One can easily show that $Convolution(Bdr(-A), Bdr(B)) = \bigcup_{K_B \infty G_{-A}} Convolution(G(K_B, G_{-A}), K(K_B, G_{-A}))$. Hence, we only need to consider *compatible* pairs to generate the *Convolution*. The computation of compatible pairs is discussed in §4.4 .

When (K_B, N_{K_B}) and $(G_{-A}, N_{G_{-A}})$ are compatible with at least one of K_B or G_{-A} being a vertex, the *Convolution* generation is especially easy, i.e. $Convolution(G_{-A}, K_B) = K_B + G_{-A}$. Let $Ch(p)$ = the *characteristic set* of $p = \{\bar{p} = p + q \mid N(B,p) \cap N(-A,q) \neq \emptyset\}$. $Ch(E) = \bigcup_{p \in E} Ch(p)$ is called the *characteristic set* of E, and $Ch(F) = \bigcup_{p \in F} Ch(p)$ is called the *characteristic set* of F. One can easily show that $Convolution(Bdr(-A), Bdr(B)) = (\bigcup_{F \in \Gamma_1} Ch(F)) \cup (\bigcup_{E \in \Gamma_2} Ch(E)) \cup (\bigcup_{p \in \Gamma_3} Ch(p))$, where Γ_1 is the set of all faces of $Bdr(B)$, Γ_2 is the set of all edges of $Bdr(B)$, and Γ_3 is the set of all vertices of $Bdr(B)$.

Generating Faces

For a face $F \subset Bdr(B)$, one can easily show that $Ch(F) = (\bigcup_{F' \infty F} Convolution(G(F,F'), K(F,F'))) \cup (\bigcup_{E \infty F} Convolution(G(F,E), K(F,E))) \cup (\bigcup_{q \infty F} Convolution(q, K(F,q)))$. One can use §4.1 to compute $Convolution(G(F,F'), K(F,F'))$ and §4.2 to compute $Convolution(G(F,E), K(F,E))$, while directly computing $Convolution(G(F,q), K(F,q)) = K(F,q) + \{q\}$ as simply translated edge segments.

Generating Edges

For an edge $E \in Bdr(B)$, one can easily show that $Ch(E) = (\bigcup_{F \infty E} Convolution(G(E,F), K(E,F))) \cup (\bigcup_{E' \infty E} Convolution(G(E,E'), K(E,E'))) \cup (\bigcup_{q \infty E} Convolution(q, K(E,q)))$. One can use §4.2 to compute $Convolution(G(E,F), K(E,F))$, and §4.3 to compute $Convolution(G(E,E'), K(E,E'))$, while directly computing $Convolution(q, K(E,q)) = \{q\} + K(E,q)$ as simply translated edge segments.

Generating Vertices

For a vertex $p \in Bdr(B)$, one can easily show that $Ch(p) = (\bigcup_{F \infty p} Convolution(G(p,F),p)) \cup (\bigcup_{E \infty p} Convolution(G(p,E),p)) \cup (\bigcup_{q \infty p} Convolution(q,p))$. Since one has $Convolution(G(p,F),p) = G(p,F) + \{p\}$, $Convolution(G(p,E),p) = G(p,E) + \{p\}$, and $Convolution(q,p) = \{q+p\}$, computing $Ch(p)$ is easy.

Note: (1) For a singular edge E and a singular vertex p the convolution edge $Convolution(G(p,E),p) = G(p,E) + \{p\}$ is a singular edge, and (2) for singular vertices p and q the convolution vertex $Convolution(q,p) = \{q+p\}$ is a singular vertex. As we will see in §4.3, (3) we can also have a singular convolution edge $Convolution(E_{-A}, E_B)$ for *parallel* edge pair E_{-A} and E_B. These are all the singularities we can have on the *C-space* obstacle boundary. As we see in this classification, all the singularities on the *C-space* obstacle boundary result from very special relations between the singularities of A and B. Most of the singularities of A and B are removed while generating the *C-space* boundary.

In §4.1–4.3 we consider both the implicit and rational parametric representation of surface patches since not all algebraic curves and surfaces have rational parametrization, see Walker (1978). For the class of *rational* algebraic curves and surfaces (which have a rational parametric form), algebraic algorithms also exist for converting between the implicit and parametric representations. However their efficiency are limited to curves and surfaces of low degree, see Abhyankar and Bajaj (1987, 1986a, b).

4.1. Generating $Convolution(F_{-A}, F_B)$

In this section, we consider how to generate the algebraic surface equation and boundary conditions of a convolution surface $Convolution(F_{-A}, F_B)$. We can use Theorem 4.1 for the case of F_{-A} and F_B being implicitly defined algebraic surfaces. Corollary 4.1 is useful when F_{-A} is implicit and F_B is parametric, or the other way around. Corollary 4.2 is useful when both F_{-A} and F_B are parametrically defined.

Theorem 4.1 : Let $F_B \subset Bdr(B)$ be a patch of an algebraic surface $f = 0$ with gradients ∇f. Further let $F_{-A} \subset Bdr(-A)$ be a patch of an algebraic surface $g = 0$ with gradients ∇g, and suppose that $(F_B, N(B, F_B))$ and $(F_{-A}, N(-A, F_{-A}))$ are *compatible*. Then $Convolution(F_{-A}, F_B)$ is the set of points $\bar{p} = (\bar{x}, \bar{y}, \bar{z}) = p + q = (x+\alpha, y+\beta, z+\gamma)$ such that

$$\begin{cases} f(x,y,z) = 0 \text{ and } p = (x,y,z) \in F_B & (1) \\ g(\alpha,\beta,\gamma) = 0 \text{ and } q = (\alpha,\beta,\gamma) \in F_{-A} & (2) \\ \nabla f \times \nabla g = 0 & (3) \\ \nabla f \cdot \nabla g > 0 & (4) \end{cases}$$

Proof : (3)–(4) are equivalent to the outward normal direction of B at p to be the same as that of $-A$ at q. □

We use Theorem 4.1 as follows. First substitute $x = \bar{x} - \alpha$, $y = \bar{y} - \beta$ and $z = \bar{z} - \gamma$ in the above equations (1) and (3). Then one can obtain the implicit algebraic equation of the $Convolution(F_{-A}, F_B)$ in terms of $\bar{x}$, $\bar{y}$ and $\bar{z}$ by eliminating α, β and γ from the equations (1)–(3). The vector equation $\nabla f \times \nabla g = 0$ gives 3 scalar equations. Since one of these equations is redundant, we can have 2 independent scalar equations from (3). Hence, we have 4 equations and eliminate 3 variables α, β, γ to get an implicit equation in terms of $\bar{x}$, $\bar{y}$, $\bar{z}$. Elimination of variables can be performed by computing resultants on pairs of equations, see Collins (1971) or the more direct method of Dixon (1908) applicable in certain cases. Eliminating variables by considering pairs of equations in general may lead to extraneous factors and special care needs to be taken in performing this step. A closed form resultant for $n-1$ variables with n equations is as yet unknown for $n \geq 3$ and is a major unsolved problem of algebraic geometry, see Abhyankar (1976).

Corollary 4.1 : Let $F_B \subset Bdr(B)$ be a patch of an algebraic surface $f=0$ with gradients ∇f. Further let $F_{-A} \subset Bdr(-A)$ be a parametric surface patch $G(u,v)=(\alpha(u,v), \beta(u,v), \gamma(u,v))$ with gradients $G_u \times G_v$, and suppose that $(F_B, N(B,F_B))$ and $(F_{-A}, N(-A,F_{-A}))$ are *compatible*. Then $Convolution(F_{-A}, F_B)$ is the set of points $\bar{p} = (\bar{x}, \bar{y}, \bar{z}) = p+q = (x+\alpha(u,v),\ y+\beta(u,v),\ z+\gamma(u,v))$ such that

$$\begin{cases} f(x,y,z)=0 \text{ and } p=(x,y,z)\in F_B & (1) \\ q=(\alpha(u,v), \beta(u,v), \gamma(u,v)) \in F_{-A} & (2) \\ \nabla f \times (G_u \times G_v) = 0 & (3) \\ \nabla f \cdot (G_u \times G_v) > 0 & (4) \end{cases}$$

First substitute $x=\bar{x}-\alpha(u,v)$, $y=\bar{y}-\beta(u,v)$ and $z=\bar{z}-\gamma(u,v)$ in the above equations (1) and (3). Then one can obtain the implicit algebraic equation of the $Convolution(F_{-A}, F_B)$ in terms of $\bar{x}$, $\bar{y}$ and $\bar{z}$ by eliminating u and v from the equations (1) and (3) by computing resultants. Since (3) gives 2 independent scalar equations, we have 3 equations and eliminate 2 variables u, v to get an implicit equation.

The case of F_B being a parametric surface and F_{-A} being an algebraic surface is similar to Corollary 4.1.

Corollary 4.2 : Let $F_B \subset Bdr(B)$ be a parametric surface patch $F(s,t)=(x(s,t), y(s,t), z(s,t))$ with gradients $F_s \times F_t$. Further let $F_{-A} \subset Bdr(-A)$ be a parametric surface patch $G(u,v)=(\alpha(u,v), \beta(u,v), \gamma(u,v))$ with gradients $G_u \times G_v$, and suppose that $(F_B, N(B,F_B))$ and $(F_{-A}, N(-A,F_{-A}))$ are *compatible*. Then $Convolution(F_{-A}, F_B)$ is the set of points $\bar{p} = (\bar{x}, \bar{y}, \bar{z}) = p+q = (x(s,t)+\alpha(u,v),\ y(s,t)+\beta(u,v),\ z(s,t)+\gamma(u,v))$ such that

$$\begin{cases} p=(x(s,t), y(s,t), z(s,t))\in F_B & (1) \\ q=(\alpha(u,v), \beta(u,v), \gamma(u,v)) \in F_{-A} & (2) \\ (F_s \times F_t) \times (G_u \times G_v) = 0 & (3) \\ (F_s \times F_t) \cdot (G_u \times G_v) > 0 & (4) \end{cases}$$

One can obtain the implicit algebraic equation of the $Convolution(F_{-A}, F_B)$ by eliminating s, t, u and v from the equations $\bar{x}=x(s,t)+\alpha(u,v)$, $\bar{y}=y(s,t)+\beta(u,v)$, $\bar{z}=z(s,t)+\gamma(u,v)$ and the above equation (3). Since (3) gives 2 independent scalar equations, we have 5 equations and need to eliminate 4 variables s, t, u, v to get an implicit equation. Further details and a time complexity analysis is given in Bajaj and Kim (1986).

4.2. Generating $Convolution(E_{-A}, F_B)$ and $Convolution(F_{-A}, E_B)$

In this section, we consider how to generate the algebraic surface equations and boundary conditions of convolution surfaces $Convolution(E_{-A}, F_B)$ and $Convolution(F_{-A}, E_B)$. We can use Theorem 4.2 for the case of E_{-A} being defined by the intersection of two implicit algebraic surfaces and F_B being an implicit algebraic surface. The other combinations of implicit and parametric surfaces defining E_{-A} and F_B have similar results as easy Corollaries of Theorem 4.2. Similar results hold for generating $Convolution(F_{-A}, E_B)$.

Theorem 4.2 : Let $F_B \subset Bdr(B)$ be a patch of an algebraic surface $f=0$ with gradients ∇f. Further let $E_{-A} \subset Bdr(-A)$ be a segment of the common edge of two faces G_{-A} and H_{-A}, where G_{-A} and $H_{-A} \subset Bdr(-A)$ are patches of algebraic surfaces $g=0$ with gradients ∇g and $h=0$ with gradients ∇h. Suppose that $(F_B, N(B,F_B))$ and $(E_{-A}, N_{E_{-A}})$ are *compatible* with $\nabla g \times \nabla h \neq 0$ on E_{-A}. Then $Convolution(E_{-A}, F_B)$ is the set of points $\bar{p} = (\bar{x}, \bar{y}, \bar{z}) = p+q = (x+\alpha,\ y+\beta,\ z+\gamma)$ such that

$$\begin{cases} f(x,y,z)=0 \text{ and } p=(x,y,z)\in F_B & (1) \\ g(\alpha,\beta,\gamma)=h(\alpha,\beta,\gamma)=0 \text{ and } q=(\alpha,\beta,\gamma)\in E_{-A} & (2) \\ \nabla f \cdot (\nabla g \times \nabla h) = 0 \text{ and } \dfrac{\nabla f}{\|\nabla f\|} \in N_{E_{-A}} & (3) \end{cases}$$

Proof : (3) is equivalent to an outward normal direction of B at p to be the same as one of the outward normal directions of $-A$ at q. □

One can obtain the implicit algebraic equation of the $Convolution(E_{-A}, F_B)$ in a similar way as in Theorem 4.1. When the face F_B is a parametric surface patch $F(s,t)=(x(s,t), y(s,t), z(s,t))$ with gradients $F_s \times F_t$, one can obtain the corresponding Corollary by changing every ∇f into $F_s \times F_t$ and the statement "$f(x,y,z)=0$ and $p=(x,y,z)\in F_B$" into "$p=(x(s,t), y(s,t), z(s,t))\in F_B$" in the above Theorem. One can make similar changes to get corresponding Corollaries for the case of G_{-A} and/or H_{-A} being parametric surface patches.

When two faces G_{-A} and H_{-A} are tangent to each other along E_{-A}, $Convolution(E_{-A}, F_B)$ is a degenerate curve on the C-*space* obstacle boundary. Actually, it is a common edge of two convolution faces generated in § 4.1.

4.3. Generating $Convolution(E_{-A}, E_B)$

In this section, we consider how to generate the algebraic surface equations and boundary conditions of a convolution surface $Convolution(E_{-A}, E_B)$. We can use Theorem 4.3 for the case of both E_{-A} and E_B being defined by two implicit algebraic surfaces. The other combinations of implicit and parametric surfaces defining E_{-A} and E_B have similar results as easy Corollaries of Theorem 4.3.

Theorem 4.3 : Let $E_B \subset Bdr(B)$ be a segment of the common edge of two faces F_B and G_B, where F_B and $G_B \subset Bdr(B)$ are patches of algebraic surfaces $f=0$ with gradients ∇f and $g=0$ with gradients ∇g. Further let $E_{-A} \subset Bdr(-A)$ be a segment of the common edge of two faces H_{-A} and K_{-A}, where H_{-A} and $K_{-A} \subset Bdr(-A)$ are patches of algebraic surfaces $h=0$ with gradients ∇h and $k=0$ with gradients ∇k. Suppose that (E_B, N_{E_B}) and $(E_{-A}, N_{E_{-A}})$ are compatible with $\nabla f \times \nabla g \neq 0$ on E_B and $\nabla h \times \nabla k \neq 0$ on E_{-A}. Then $Convolution(E_{-A}, E_B)$ is the set of points $\bar{p} = (\bar{x}, \bar{y}, \bar{z}) = p+q = (x+\alpha,\ y+\beta,\ z+\gamma)$ such that

$$\begin{cases} f(x,y,z)=g(x,y,z)=0 \text{ and } p=(x,y,z)\in E_B & (1) \\ h(\alpha,\beta,\gamma)=k(\alpha,\beta,\gamma)=0 \text{ and } q=(\alpha,\beta,\gamma)\in E_{-A} & (2) \\ \dfrac{\lambda\cdot\nabla f+(1-\lambda)\cdot\nabla g}{\|\lambda\cdot\nabla f+(1-\lambda)\cdot\nabla g\|} \in N_{E_{-A}} \text{ and} & \\ \dfrac{\mu\cdot\nabla h+(1-\mu)\cdot\nabla k}{\|\mu\cdot\nabla h+(1-\mu)\cdot\nabla k\|} \in N_{E_B} \text{ for } some\ 0\leq\lambda,\mu\leq 1 & (3) \end{cases}$$

Proof : (3) is equivalent to an outward normal direction of B at p to be the same as an outward normal direction of $-A$ at q. □

One can obtain the implicit algebraic equation of the $Convolution(E_{-A}, E_B)$ in a similar way as in Theorem 4.1.

When F_B and G_B are tangent to each other along E_B, or H_{-A} and K_{-A} are tangent to each other along E_{-A}, $Convolution(E_{-A}, E_B)$ is a degenerate curve on the C-*space* obstacle boundary and is a common edge of two convolution faces generated in § 4.2. In the special case of F_B and G_B being tangent along E_B, and also H_{-A} and K_{-A} being tangent along E_{-A}, $Convolution(E_{-A}, E_B)$ is a degenerate point.

Let $N_{E_s}(p) = N_{E_s} \cap N(S,p)$, then $N_{E_s}(p)$ is a geodesic arc on S^2. When two line segments in a plane intersects, either there is a unique intersection point or they overlap entirely on the same line. One can show a similar fact for minimal geodesic arcs on S^2 as follows.

Fact 4.1 : If $N_{E_s}(p) \cap N_{E_{-A}}(q) \neq \emptyset$, either (1) $N_{E_s}(p) \cap N_{E_{-A}}(q)$ is a point or (2) $N_{E_s}(p) = N_{E_{-A}}(q)$.

By subdividing E_B and E_{-A} if necessary, we may assume only one of the conditions (1) or (2) holds for the whole edges E_B and E_{-A}. We call E_B and E_{-A} to be *parallel* if the condition (2) holds on the whole edges E_B and E_{-A}. If E_B and E_{-A} is a *parallel* edge pair, the *Convolution* (E_{-A}, E_B) generated in Theorem 4.3 is a degenerate curve on the *C-space* obstacle. Otherwise it is a surface patch.

5. Obtaining Gaussian Model of *C-space* Obstacles

We now show how to construct the Gaussian (Spherical) Model of $CO(A,B)$, see Figures 2 (a)–(c). Let $S^2{}_B$ and $S^2{}_{-A}$ be the Gaussian Models of B and $-A$. These define graphs on S^2 with degeneracies tagged appropriately. Let a new graph $S^2{}_{CO(A,B)}$ on S^2 be the overlay of $S^2{}_B$ and $S^2{}_{-A}$. Then $S^2{}_{CO(A,B)}$ is the Gaussian Model of $CO(A,B)$ and determines all *compatible* face, edge and vertex pairs between $Bdr(B)$ and $Bdr(-A)$. Further the topology of the $Bdr(CO(A,B))$ is given by the topology of the tesselated $S^2{}_{CO(A,B)}$ surface. Construction of $S^2{}_{CO(A,B)}$ requires computing the intersections of edges of $S^2{}_B$ with edges of $S^2{}_{-A}$. These intersections can be computed by using the following Theorems. The intersection of two geodesic arcs can be computed by Theorem 5.1. The intersection of one general curve segment and one geodesic arc can be computed by Theorem 5.2. The intersection of two general curve segments can be computed by Theorem 5.3.

Next by using a spherical sweep algorithm where one can move a great circle around the sphere and amongst the edge segments, it is possible to compute all the overlay curve intersections. The details are somewhat intricate but a straightforward generalization of moving a line in a plane-sweep algorithm.

Theorem 5.1 : Let γ be a geodesic arc connecting n_1 to n_2 on $S^2{}_B$ and γ' be a geodesic arc connecting n'_1 to n'_2 on $S^2{}_{-A}$. Then γ and γ' intersect at $(\lambda \cdot n_1 + (1-\lambda)\cdot n_2) \,/\, \|\lambda \cdot n_1 + (1-\lambda)\cdot n_2\|$ if and only if

$$\begin{cases} (\lambda \cdot n_1 + (1-\lambda)\cdot n_2) \times (\mu \cdot n'_1 + (1-\mu)\cdot n'_2) = 0 & (1) \\ (\lambda \cdot n_1 + (1-\lambda)\cdot n_2) \cdot (\mu \cdot n'_1 + (1-\mu)\cdot n'_2) > 0 & (2) \end{cases}$$

for some $0 \le \lambda,\ \mu \le 1$.

Proof : These two conditions are equivalent to that $\lambda \cdot n_1 + (1-\lambda)\cdot n_2$ is in the same direction as $\mu \cdot \bar{n}_1 + (1-\mu)\cdot \bar{n}_2$ for some $0 \le \lambda,\ \mu \le 1$.

Since the vector equation (1) gives two independent scalar equations in two variables λ, μ, one can solve this system of polynomial equations either numerically or symbolically by using polynomial resultants or Grobner Bases, see Buchberger, Collins, and Loos (1982).

Theorem 5.2 : Let γ be a curve segment on $S^2{}_B$ given by the set of points $\nabla f(p) / \|\nabla f(p)\|$ for $p \in E$, where $E \subset Bdr(B)$ is a segment of the common edge of two faces F and G both being patches of algebraic surfaces $f = 0$ with gradients ∇f and $g = 0$ with gradients ∇g. And, let γ' be a geodesic arc connecting n_1 to n_2 on $S^2{}_{-A}$. Then γ and γ' intersect at $\nabla f(p) / \|\nabla f(p)\|$ if and only if

$$\begin{cases} f(x,y,z) = g(x,y,z) = 0 \ \text{ and } \ p = (x,y,z) \in E & (1) \\ \nabla f \cdot (n_1 \times n_2) = 0 & (2) \\ \nabla f \cdot (n_1 - (n_1 \cdot n_2) n_1) > 0 & (3) \\ \nabla f \cdot (n_2 - (n_2 \cdot n_1) n_2) > 0 & (4) \end{cases}$$

Proof : (2)–(4) are equivalent to that ∇f is in the same direction as $\lambda \cdot n_1 + (1-\lambda)\cdot n_2$ for some $0 \le \lambda \le 1$.

Since (1)–(2) give three equations in three variables x, y, z, one can solve this system of polynomial equations.

Theorem 5.3 : Let γ be a curve segment on $S^2{}_B$ given by the set of points $\nabla f(p) / \|\nabla f(p)\|$ for $p \in E_B$, where $E_B \subset Bdr(B)$ is a segment of the common edge of two faces F and G both being patches of algebraic surfaces $f = 0$ with gradients ∇f and $g = 0$ with gradients ∇g. And, let γ' be a curve segment on $S^2{}_{-A}$ given by the set of points $\nabla h(q) / \|\nabla h(q)\|$ for $q \in E_{-A}$, where $E_{-A} \subset Bdr(-A)$ is a segment of the common edge of two faces H and K both being patches of algebraic surfaces $h = 0$ with gradients ∇h and $k = 0$ with gradients ∇k. Then γ and γ' intersect at $\nabla f(p) / \|\nabla f(p)\|$ if and only if

$$\begin{cases} f(x,y,z) = g(x,y,z) = 0 \ \text{ and } \ p = (x,y,z) \in E_B & (1) \\ h(\alpha,\beta,\gamma) = k(\alpha,\beta,\gamma) = 0 \ \text{ and } \ q = (\alpha,\beta,\gamma) \in E_A & (2) \\ \nabla f \times \nabla h = 0 & (3) \\ \nabla f \cdot \nabla h > 0 & (4) \end{cases}$$

Proof : (3)–(4) are equivalent to that ∇f is in the same direction as ∇h.

Since the vector equation (3) gives two independent scalar equations, one has 6 scalar equations in 6 variables from (1)–(3) and can solve this system of polynomial equations.

Each face of the subdivision $S^2{}_{CO(A,B)}$ corresponds to a compatible pair $((K_B, N_{K_B}), (G_{-A}, N_{G_{-A}}))$ of boundary elements of $Bdr(B)$ and $Bdr(-A)$. Note that we consider the degenerate curves and degenerate points as generic faces of $S^2{}_C$. Using the formula defining K_B and G_{-A} one can compute the equation for *Convolution* (G_{-A}, K_B). The boundary curves of surface patchs or the end points of edges for *Convolution* $(Bdr(B), Bdr(-A))$ can be computed by using the boundary informations given by N_{K_B} $(= N_{G_{-A}})$. In various applications we do not need to construct the entire *C-space* obstacle boundary or the entire Gaussian Model $S^2{}_{CO(A,B)}$. For example, consider the compliant motion which we will describe in § 6. In the first approach, one can construct only the faces of *C-space* obstacle boundary which correspond to the sequence of faces of $S^2{}_{CO(A,B)}$ along which one plans a compliant path. In the second approach, once one obtains a shortest path on the approximating convex polyhedron, one needs to generate only part of $S^2{}_{CO(A,B)}$ and *C-space* obstacle boundary faces which correspond to the approximating shortest path.

6. Compliant Motion in *C-space*

Having seen ways of constructing the curved surface boundary of the *C-space* obstacles our next step is to generate continuous curves on the *C-space* obstacle boundary connecting various specified start and end points. These correspond to continuous compliant paths for the original object and obstacles.

In the past several authors have posed solutions to finding paths between points on *C-space* obstacles bounded by planar faces. Sharir and Schorr (1984) presented an $O(n^3 \log n)$ algorithm for finding shortest paths on n vertex convex polyhedra. This was subsequently improved to $O(n^2 \log n)$, Mount (1985). O'Rourke, Suri, and Booth (1985) gave an $O(n^5)$ algorithm for finding shortest paths on non-convex polyhedra which was later improved to $O(n^2 \log n)$ in Mitchell, Mount, and Papadimitriou (1985).

Finding shortest paths on curved *C-space* obstacles with algebraic boundary surfaces as generated in the earlier section, is correspondingly more difficult. On single surfaces such paths are known as *geodesics*, and arise in cutter tool paths in machining, efficient terrain navigation and winding rotor coils, etc. There are closed solutions of geodesics for various quadrics and surfaces of revolution. For given two points p and q on a closed surface there exists a minimal geodesic joining p and q, see do Carmo (1976). However computing exact analytical solutions for geodesics in general is quite difficult since these are given as solutions of nonlinear ordinary differential equations, Beck, Farouki, and Hinds (1986). Even for certain simple algebraic surfaces the geodesic curves are non-unique and non-algebraic in nature, e.g., a non-algebraic helical curve is geodesic on a cylindrical surface (algebraic degree two).

6.1. Arbitrary Compliant Paths

Before considering shortest paths on curved convex *C-space* obstacle boundaries however we first consider the generation of certain algebraic curves which lie on the boundary and connect specified start points (s_i) and end points (e_j) on the boundary. Such curves provide compliant paths for a certain object and obstacle pair. Perhaps the simplest method is to choose a point p in the interior of the convex *C-space* obstacle and consider the unique plane PL containing a specified pair s_i and e_j, and the interior point p. The intersection of PL with the convex *C-space* obstacle boundary yields an algebraic curve on the boundary with breakpoints on vertices and edges. The correctness of this procedure is justified by the following Fact.

Fact 6.1: A plane passing through an interior point of a convex object intersects the convex object boundary in a planar convex curve which is a Jordan curve (i.e. a curve homeomorphic to a circle).

The disadvantage of this simplistic approach is that the arbitrary choice of the interior point p may yield undesirable algebraic curves which consist of complete singular edges and also pass through various singular vertices of the *C-space* obstacle boundary. Such paths then correspond to vertices and edges of the object riding on vertices and edges of the obstacle during compliant motion, see path γ_1 in Figure 3.

To circumvent this undesirable prospect an alternate method for choosing compliant paths may be adopted as follows. Consider the *Gaussian* (Spherical) Model of the *C-space* obstacle as constructed in §5. The start and end points of the *C-space* obstacle surface are easily mapped to this spherical model. Next piecewise geodesic curves on the sphere can be constructed connecting these start and end points which avoid degeneracies on the *C-space* obstacle boundary. Various interior points to surface patches on the sphere can be chosen and for each triple of points (two surface points and the center of sphere) the corresponding plane intersection with the sphere provide geodesic curve segments on the sphere, see path γ_2 in Figure 3. These piecewise curves on the sphere are then mapped to algebraic curves on the *C-space* obstacle boundary. In this way compliant paths which are piecewise algebraic can be determined which traverse a specific desired sequence of surface patches or desired sequence of surface and edge contacts between object and obstacle.

6.2. Geodesic Approximation

We now describe a method of obtaining piecewise algebraic approximate geodesic paths on convex *C-space* obstacles with algebraic surfaces. One first computes a "good" approximate convex polyhedron of the curved convex *C-space* obstacle as we now describe. Next the start and end points on the *C-space* obstacle boundary are mapped onto the convex polyhedron and the shortest path between these points is computed on this polyhedron by using say the $O(n^2 \log n)$ algorithm, Mount (1985). Finally the approximate shortest path of the *C-space* obstacle boundary $Bdr(CO(A,B))$ can be obtained by projecting the shortest path of the polyhedron onto $Bdr(CO(A,B))$.

Hierarchical Convex Polyhedral Approximation

Suppose S is strictly convex, i.e. S is convex and for each point $p \in Bdr(S)$ the supporting plane L_p has only one common point (i.e. p) with S. Strictly convex objects exclude degeneracies like ruled surface patches, planar patches, etc. We consider the strictly convex case first and then consider the general convex case by adding special features to the strictly convex case. The following Theorems hold for the boundaries of strictly convex objects.

Theorem 6.1 : Let $F \subset Bdr(S)$ be a patch of an algebraic surface $f = 0$ with gradients ∇f. Suppose $\{e\} = N(S,p)$ for some $p \in F$, then $p = (x,y,z)$ is the solution of the following equations.

$$\begin{cases} f(x,y,z) = 0 \text{ and } p = (x,y,z) \in F & (1) \\ \nabla f \times e = 0 & (2) \\ \nabla f \cdot e > 0 & (3) \end{cases}$$

Proof : (2)–(3) are equivalent to that ∇f is in the same direction as e. □

Theorem 6.2 : Let (E, N_E) be a pair such that $E \subset Bdr(S)$ be a segment of the common edge of two faces F and G, where F and $G \subset Bdr(S)$ are patches of algebraic surfaces $f = 0$ with gradients ∇f and $g = 0$ with gradients ∇g, and $N_E \subset N(S,E)$ with $N_E \cap N(S,p) \neq \varnothing$ for all $p \in E$. Suppose $e \in N_E \cap N(S,p)$ for some $p \in E$, then $p = (x,y,z)$ is the solution of the following equations.

$$\begin{cases} f(x,y,z) = g(x,y,z) = 0 \text{ and } p = (x,y,z) \in E & (1) \\ e \cdot (\nabla f \times \nabla g) = 0 \text{ and } e \in N_E & (2) \\ e \cdot ((\nabla g \cdot \nabla g)\nabla f - (\nabla f \cdot \nabla g)\nabla g) > 0 & (3) \\ e \cdot ((\nabla f \cdot \nabla f)\nabla g - (\nabla f \cdot \nabla g)\nabla f) > 0 & (4) \end{cases}$$

Proof : (2)–(4) are equivalent to that $e \in N_E \cap N(S,p)$. □

When F is a parametric surface patch $F(s,t) = (x(s,t), y(s,t), z(s,t))$ with gradients $F_s \times F_t$, one can obtain the corresponding Corollaries by changing ∇f into $F_s \times F_t$ and the statement "$f(x,y,z)=0$ and $p=(x,y,z) \in F$" into "$p = (x(s,t), y(s,t), z(s,t)) \in F$" in the above Theorems. Similarly in the case of G being a parametric surface patch. Note the simultaneous solution of systems of polynomial equations can be solved either numerically or symbolically by using polynomial resultants, see also Buchberger, Collins, and Loos (1982).

Using the above Theorems we can approximate a strictly convex object S by convex polyhedra as follows. Our approximation scheme is hierarchical and curvature dependent. At the coarsest level we inscribe a regular polyhedron (say, icosahedron) inside a unit sphere and project it onto the surface of the sphere, see Figure 4 (a)–(b). This projection defines a regular subdivision or tesselation on S^2. We can further triangulate each polyhedral face if it is not triangular. Let $S^2{}_T$ be this triangular subdivision of S^2, and $S^2{}_S$ be the subdivision given by the Gaussian Model of S. Further let $S^2{}_{T_s}$ be the overlay of $S^2{}_T$ and $S^2{}_S$. The overlay $S^2{}_{T_s}$ can be computed by using Theorems 5.1–5.2. To compute an approximating polyhedron of S corresponding to the triangulation S_T one computes the boundary points of S at which $Bdr(S)$ have gradients corresponding to vertices of S_T. For each vertex e of $S^2{}_T$, by looking at the vertex e in $S^2{}_{T_s}$ one can tell which face, edge or vertex of $Bdr(S)$ has e as one of its outward normal direction(s). Suppose a face F with a surface equation $f = 0$ with gradients ∇f has e as one of its outward normal directions at p, then the point p can be computed by using Theorem 6.1. When an edge E whose adjacent faces are given by $f = 0$ and $g = 0$ with gradients ∇f and ∇g, has e as one of its outward normal directions at p, then the point p can be computed by using Theorem 6.2. If e is in the Gaussian Image of a vertex p, the point p is directly obtained by the coordinates of p. When any of the faces are parametric surfaces, one can use a modified version of Theorem 6.1 or 6.2 as discussed before.

Further we can approximate S by two related polyhedra V and W (an inner and an outer) determined by the points p corresponding to the vertices e of S^2_T. Let P be the set of all these points p. Since S is convex, V = *Convex−Hull* $(P) \subset S$. Let L_p be the supporting plane of S at p and H_p = the half-space defined by L_p such that $S \subset H_p$, then $S \subset W = \bigcap_{p \in P} H_p$. For a strictly convex object S each face of V is a triangle F_{ijk} determined by three boundary points p_i, p_j and p_k such that the corresponding Gaussian Images e_i, e_j and e_k makes a triangular face N_{ijk} on the tessellation of the Gaussian Sphere. F_{ijk} is a planar approximation of the boundary surface patch S_{ijk} on which S has normal directions corresponding to angular range N_{ijk}. Each vertex of W is a common intersection point H_{ijk} of three supporting planes L_{p_i}, L_{p_j} and L_{p_k}. The distance between the vertex H_{ijk} and the face F_{ijk} gives an estimation of how closely V and W approximates S over S_{ijk}. If the difference is bigger than a predefined bound (say, $\varepsilon > 0$), we can further triangulate N_{ijk} into a finer resolution and modify V and W locally by repeating the above procedure. We can continue this refinement over all coarsely approximated faces and recursively to all the sub faces thus obtained, see Figure 4 (c).

When a convex object S is not strictly convex, $Bdr(S)$ can have some ruled surface pathes and planar patches which have degenerate curves and points as Gaussian Images. The degenerate point for the Gaussian Image of a planar patch can be on the interior of a single triangular patch, on the common edge of two triangular patches, or at a common vertex of several triangular patches. Polygonal approximation can be obtained by connecting the vertices of the planar patch and the vertices of the triangular patches where the degenerate point lies. The degenerate curve for the Gaussian Image of a ruled surface patch passes through a sequence of triangular patches of the Gaussian Sphere. By subdividing this curve into finite subsegments we can approximate the ruled surface into a finite sequence of planar patches. Each corner of this planar patches can be connected to the vertices of the triangular patches where the segments of the degenerate curve lie.

Approximating Shortest Path

By continuing the above finer refinements upto an arbitrarily small $\varepsilon > 0$, we can construct a sequence $\{V_n\}$ of inscribed convex polyhedra converging to S. It can be shown that the areas of $Bdr(V_n)$ converges to those of $Bdr(S)$. Hence, this fact justifies our strategy of using the shortest paths of $Bdr(V_n)$ as approximate shortest paths of $Bdr(S)$ for sufficiently large n.

Convex Polyhedral Approximation of *C−space* Obstacle

One may compute a convex polyhedral approximation of $CO(A,B)$ by applying the above procedure to the boundary representation and the Gaussian Model of $CO(A,B)$. But, since the degrees of *C−space* obstacle boundary surfaces are extremely high, it is more efficient to deal with the boundary surfaces of $-A$ and B to approximate $CO(A,B)$. One can start with a triangular subdivision S^2_T as before. But, at this time, instead of computing the point $p \in Bdr(CO(A,B))$ which has an outward normal direction e on the *C−space* obstacle boundary, one can compute $p_B \in Bdr(B)$ and $p_{-A} \in Bdr(-A)$ which correspond to e on the object and obstacle boundary surface, and compute $p_B + p_{-A}$. One can continue the refinement steps as above.

In this way we do not need to even construct the entire Gaussian Model of $CO(A,B)$ either. Computation of p_B and p_{-A} requires computing the overlays of S^2_T with S^2_B and S^2_{-A}. Only a partial boundary of $S^2_{CO(A,B)}$ needs to be constructed to obtain the topology of the corresponding partial boundary of $Bdr(CO(A,B))$ where the approximate shortest path lies.

Shortest Path on Convex Polyhedron and Projection onto Curved *C−space* Obstacle

Once we have a convex polyhedron V approximating $CO(A,B)$ we can compute a shortest path γ on $Bdr(V)$ by using say the $O(n^2 \log n)$ algorithm, Mount (1985). By projecting γ from an interior point onto $Bdr(CO(A,B))$ one can get an approximate shortest path on the *C−space* obstacle boundary. One can choose the center of mass c of V as the projection point and consider the intersections of the plane (containing each line segment of the shortest path on V and c) with the corresponding surface patch of $Bdr(CO(A,B))$. This surface patch corresponds to the triangular facet of V which contains the line segment which we are projecting. For certain applications with complicated *C−space* obstacles, a single projection point may not give nice properties for the projected curve. We may have this difficulty for instance when the *C−space* obstacle has many degenerate boundary surfaces. In this case, we can choose a variety of projection points or use other projection methods, whichever is appropriate for the application.

7. Conclusion

We have described algorithmic methods for computing *C−space* obstacles and determining paths on their boundary, using boundary representation and Gaussian image geometric models. Many further algorithmic developments need to be made before which one can use computer models of the workspace and the objects being manipulated and simulate and reason with them. Such reasoning for instance could allow a robot system with off-line programming: to figure out a grip, to determine the stability of an assembly, to figure out a part assembly sequence and so forth.

8. References

Abhyankar, S., (1976)
Historical Ramblings in Algebraic Geometry and Related Algebra, *American Mathematical Monthly*, 83, 6, 409-448.

Abhyankar, S., and Bajaj, C., (1987)
Automatic Rational Parameterization of Curves and Surfaces I: Conics and Conicoids, *Computer Aided Design*, 19, 1, 11-14.

Abhyankar, S., and Bajaj, C., (1986a)
Automatic Rational Parameterization of Curves and Surfaces II: Cubics and Cubicoids, Computer Science Technical Report CSD-TR-592, Purdue University.

Abhyankar, S., and Bajaj, C., (1986b)
Automatic Rational Parameterization of Curves and Surfaces III: Algebraic Plane Curves, Computer Science Technical Report CSD-TR-619, Purdue University.

Adamowicz, M., and Albano, A., (1976)
Nesting two-dimensional shapes in rectangular modules, *Computer Aided Design*, 2, 1, 27-33.

Bajaj, C., and Kim, M., (1986)
Generation of Configuration Space Obstacles II: The Case of Moving Algebraic Surfaces, Computer Science Technical Report CSD-TR-586, Purdue University.

Bajaj, C., and Kim, M., (1987)
Generation of Configuration Space Obstacles III: The Case of Moving Algebraic Curves, *Proc. 1987 IEEE International Conference on Robotics and Automation*, North Carolina.

Beck, J., Farouki, R., and Hinds, J., (1986)
Surface Analysis Methods, *IEEE Comp. Graphics and Appl.*, 18-36.

Buchberger, B., Collins, G., and Loos, R.,(editors), (1982)
Computer Algebra, Symbolic and Algebraic Computation, Computing Supplementum 4, Springer Verlag, Wien New York.

Canny, J., (1984)
Collision Detection for Moving Polyhedra, A.I. Memo o. 806, MIT.

Chazelle, B., (1984)
Convex Partitions of Polyhedra: A lower bound and worst-case optimal algorithm, *SIAM J. on Computing*, 13, 3, 488-507.

Collins, G., (1971)
The Calculation of Multivariate Polynomial Resultants, *Journal of the ACM*, 18, 4, 515-532.

Dixon, A., (1908)
The Eliminant of Three Quantics in Two Independent Variables, *Proc. London Mathematical Society*, 2, 6, 468-478.

do Carmo, M., (1976)
Differential Geometry of Curves and Surfaces, Prentice Hall, Engelwood, N.J.

Donald, B., (1984)
Motion Planning with Six Degrees of Freedom A.I. Technical Report 791, MIT.

Franklin, W., and Akman, V., (1984)
Shortest Paths Between Source and Goal Points Located On/Around a Convex Polyhedron, *22nd Allerton Conference on Communication, Control and Computing*, 103-112.

Freeman, H., (1975)
On the packing of arbitrary shaped templates, *Proc. 2nd USA-Japan Computer Conference*, 102-107.

Guibas, L., Ramshaw, L., and Stolfi, J., (1983)
A Kinetic Framework for Computational Geometry, Proc. 24th Annual Symp. on Foundations of Computer Science, 100-111.

Guibas, L., and Seidel, R., (1986)
Computing Convolutions by Reciprocal Search, *Proc. of 2nd ACM Symposium on Computational Geometry*, 90-99.

Hoffmann, C., and Hopcroft, J., (1986)
Geometric Ambiguities in Boundary Representation, TR86-725, Computer Science, Cornell University.

Hopcroft, J., and Krafft, D., (1985)
The Challenge of Robotics for Computer Science, Manuscript.

Horn, B.K.P., (1986)
Robot Vision, McGraw-Hill Book Co, MIT Press.

Kelly, P., and Weiss, M., (1979)
Geometry and Convexity, John Wiley & Sons, New York.

Lozano-Perez, T., (1983)
Spatial Planning: A Configuration Space Approach, *IEEE Trans. on Computers*, v.C-32, 108-120.

Lozano-Perez, T., and Wesley, M.A., (1979)
An algorithm for planning collision free paths among polyhedral obstacles, *Communications of the ACM*, 22, 560-570.

Lozano-Perez, T., Mason, M., and Taylor, R., (1984)
Automatic Synthesis of Fine-Motion Strategies for Robots, *The Intl. J. of Robotics Research*, v. 3, 1, 3-24.

Mitchell, J., Mount, D., and Papadimitriou, C., (1985)
The Discrete Geodesic Problem, Manuscript.

Mount, D., (1985)
On Finding Shortest Paths on Convex Polyhedra, Technical Report 120, Center for Automation Research, University of Maryland.

O' Rourke, J., Suri, S., and Booth, H., (1985)
Shortest Paths on Polyhedral Surfaces, *Proc. Symposium on Theoretical Aspects of Computing*, 243 -254.

Pogorelov, A.V., (1978)
The Minkowski Multidimensional Problem, Winston, New York.

Requicha, A., (1980)
Representations of Rigid Solid Objects, *Computer Aided Design*, Springer Lecture Notes in Computer Science 89, 2-78.

Requicha, A., and Voelcker, H., (1983)
Solid Modeling: Current Status and Research Directions, *IEEE Comp. Graphics and Appl.*, 25-37.

Schwartz, J. and Sharir, M., (1983)
On the Piano Movers Problem: II, General Techniques for Computing Topological Properties of Real Algebraic Manifolds, *Adv. Appl. Math.* 4, 298-351.

Sharir, M., and Schorr, A., (1984)
On Shortest Paths in Polyhedral Spaces, *Proc. 16th ACM Symposium on Theory of Computing*, 144-153.

Tiller, W., and Hanson, E., (1984)
Offsets of Two-Dimensional Profile, *IEEE Computer Graphics & Applications*, Sept., 36-46.

Udupa, S., (1977)
Collision detection and avoidance in computer controlled manipulators, *Proc. 5th Int. Joint Conf. Artificial Intelligence*, MIT, 737-748.

Walker, R., (1978)
Algebraic Curves, Springer Verlag, New York.

Yap, C., (1985)
Algorithmic Motion Planning, Manuscript.

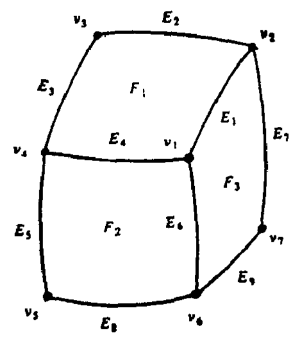

Figure 1–(a) Convex object

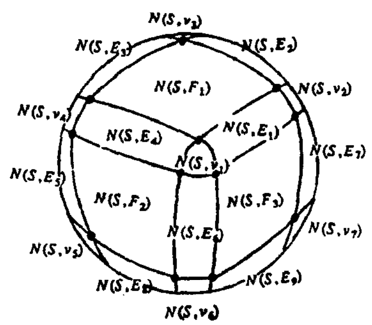

Figure 1–(b) Gaussian Model for the object (a)

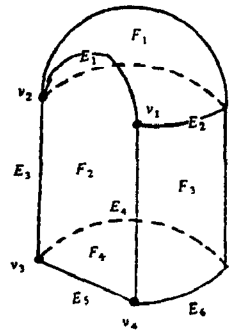

Figure 1–(c) Convex object with ruled surface patch and planar surface patch on the boundary

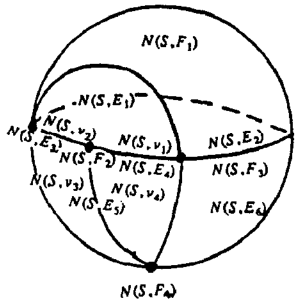

Figure 1–(d) Gaussian Model for the object (c)

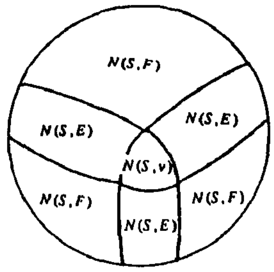

Figure 2–(a) Gaussian Model for $-A$

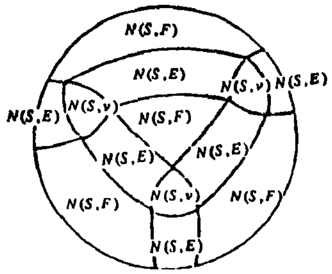

Figure 2–(b) Gaussian Model for B

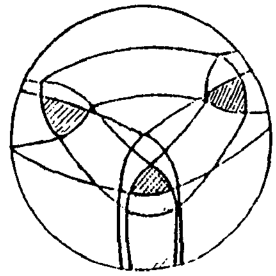

Figure 2–(c) Gaussian Model for $CO(A,B)$

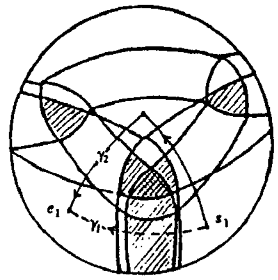

Figure 3 Compliant path on C-*space* obstacle boundary

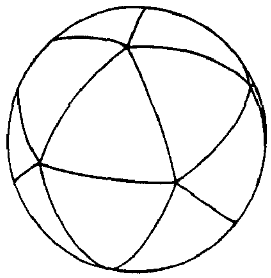

Figure 4–(b) Icosahedron projected onto the unit sphere

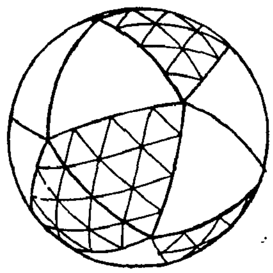

Figure 4–(c) Selective refinement

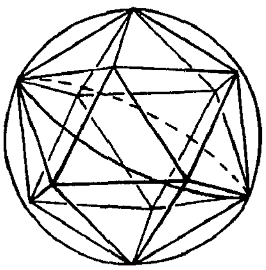

Figure 4–(a) Icosahedron inscribed in the unit sphere