Leo Joskowicz
Institute of Computer Science
The Hebrew University
Jerusalem 91904, Israel

Russell H. Taylor
Computer Science Department
The Johns Hopkins University
Baltimore, Maryland 21218

Interference-Free Insertion of a Solid Body Into a Cavity: An Algorithm and a Medical Application

Abstract

This article presents a novel algorithm for efficiently computing an interference-free insertion path of a body into a cavity and shows its practical use in the insertability analysis of custom orthopedic hip implants. The algorithm is designed to handle tightly fit, very complex three-dimensional bodies requiring fine, complex, coupled six-degree-of-freedom motions in a preferred direction. It provides a practical method for efficiently handling the geometric complexity of tight-fit insertions. The algorithm computes an insertion path consisting of small interference-free body motion steps. It formulates local, linearized configuration space constraints derived from the shapes and computes successive motion steps by solving a series of linear optimization problems whose solution corresponds to the maximum allowed displacement in a preferred direction satisfying the constraints. It either finds a successful insertion path or a stuck configuration. We demonstrate the algorithm with EXTRACT, a program for analyzing the insertability of cementless custom orthopedic hip implants. EXTRACT computes interference-free insertion paths for tightly fit implant and canal shapes described with 10,000 facets to an accuracy of 0.01 inch in 30 minutes on a workstation. It has been successfully tested on 30 real cases provided by a medical equipment manufacturer.

1. Introduction

This article presents a novel algorithm for efficiently computing an interference-free insertion path of a body into a cavity. The problem of computing such paths has been widely studied in robotics, where it is referred to as the *peg-in-hole problem.* It is an instance of the general motion planning problem in which a peg (the moving body) is to be inserted without interferences into a hole (the fixed cavity). Typically, the peg and the hole are tightly fit, so that the clearance between the peg and the hole is small; the peg and hole shapes are almost complementary; and a general, preferred insertion direction is known—the main axis of the hole. Examples of such fits include simple shapes (e.g., pins and holes, screws and bolts, fasteners and part covers), and complex shapes, (e.g., molds and prosthetic implants) (Fig. 1).

Computing interference-free insertion paths is essential for analyzing the insertability of a body into a cavity. Insertability analysis is ubiquitous in a wide variety of tasks and domains. In robotics and manufacturing assembly, it is necessary to find and execute the insertion path of a robotically guided part into a mating fixture. In engineering design, it is necessary to detect part interferences, design tightly fit machine parts, and establish tolerances for assemblability. In molding, it is necessary to verify that a mold can be removed from the part it shapes once the part has solidified. In biomedical engineering, it is necessary to design and validate prosthetic implants. Insertability analysis is required to validate shape designs; identify interferences, blocking surfaces, and stuck configurations; make shape modifications; and explore shape alternatives.

Computing insertion paths for tight fits requires limited, localized search of high geometric complexity. Body motions are highly constrained by the cavity walls, so compliant motions along the preferred direction are likely to insert the body into the cavity. However, because the shapes are complex and the clearance is small, many interference tests are necessary. When the shapes require thousands of facets to describe them to guarantee reliable results, the geometric computation complexity dominates the search time.

We have developed an insertion algorithm that effectively addresses the geometric complexity of path construction. The algorithm is designed to handle tightly fit, very complex three-dimensional bodies requiring fine, complex, coupled six-degree-of-freedom motions in a preferred direction. It emphasizes efficient local geome-

The International Journal of Robotics Research,
Vol. 15, No. 3, June 1996, pp. 211–229,

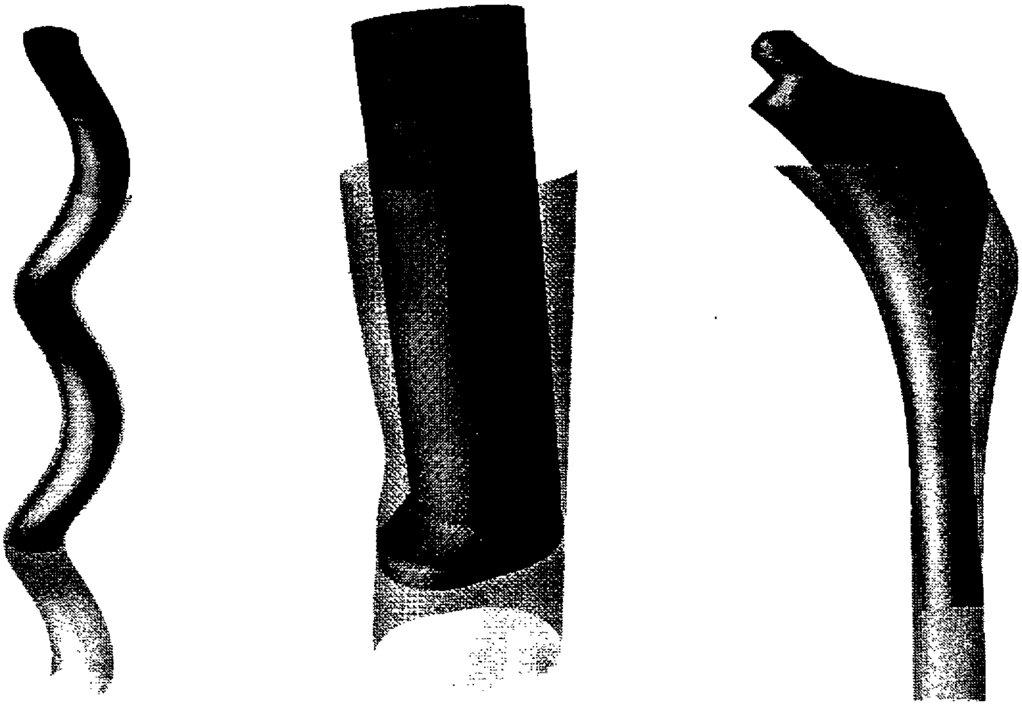

Fig. 1. Three instances of peg-in-hole problems. (A) A screw in a helical tube. (B) A boot-shaped body in a bended tube. (C) An orthopedic hip implant in a canal (the dark body is the peg, the translucid body is the hole).

try and motion constraint computation over search. The algorithm computes an insertion path consisting of small interference-free body motion steps. It formulates local, linearized configuration space constraints derived from the shapes and computes successive motion steps by solving a series of linear optimization problems whose solution corresponds to the maximum allowed displacement in a preferred direction satisfying the constraints. It finds either a successful insertion path or a stuck configuration. The algorithm implements a greedy path finding strategy with localized backtracking that produces quasi-monotone insertion paths to any desired resolution.

We demonstrate the practical use of the algorithm with EXTRACT, a program for computing insertion paths for cementless custom orthopedic hip implants into a matching cavity prepared in the patient's femur (Fig. 2). EXTRACT computes interference-free insertion paths for tightly fit implant and canal shapes described with 10,000 facets to an accuracy of 0.01 inch in 30 minutes on a workstation (Fig. 3). It has been successfully tested on 30 real cases provided by a medical equipment manufacturer.

This article describes the insertion algorithm, its implementation, and the experimental results. Section 2 reviews related work in path planning. Section 3 provides an overview of the solution. Section 4 describes the problem formulation and introduces assumptions and approximations. Section 5 describes the insertion algorithm and discusses related algorithmic issues. Section 6 describes EXTRACT, the program implementation, and its use in the insertability analysis of custom hip implants. Section 7 concludes with a summary and a discussion on possible extensions and applications. An appendix formalizes the concepts of tight fit and configuration space approximation. Readers interested in the medical application can first go directly to Section 6, read the insertion algorithm in Section 5, and optionally refer to the configuration space formulation in Sections 3 and 4.

2. Related Work

Computing an insertion path of a body into a cavity is an instance of the classic path planning problem where the goal is to find an interference-free path of one or more moving objects from an initial to a final configuration amidst fixed obstacles (Latombe 1991). Finding such a path requires searching the space of object configurations (its configuration space) for a continuous, nonoverlapping path from the initial to the final configuration. There are two main strategies for finding such paths: global strategies and local strategies. Global strategies first construct and partition the configuration space into cells, then construct its connectivity graph, and then search the graph for the desired path. Local strategies directly search for the path, performing the necessary geometric computations that guarantee nonoverlapping

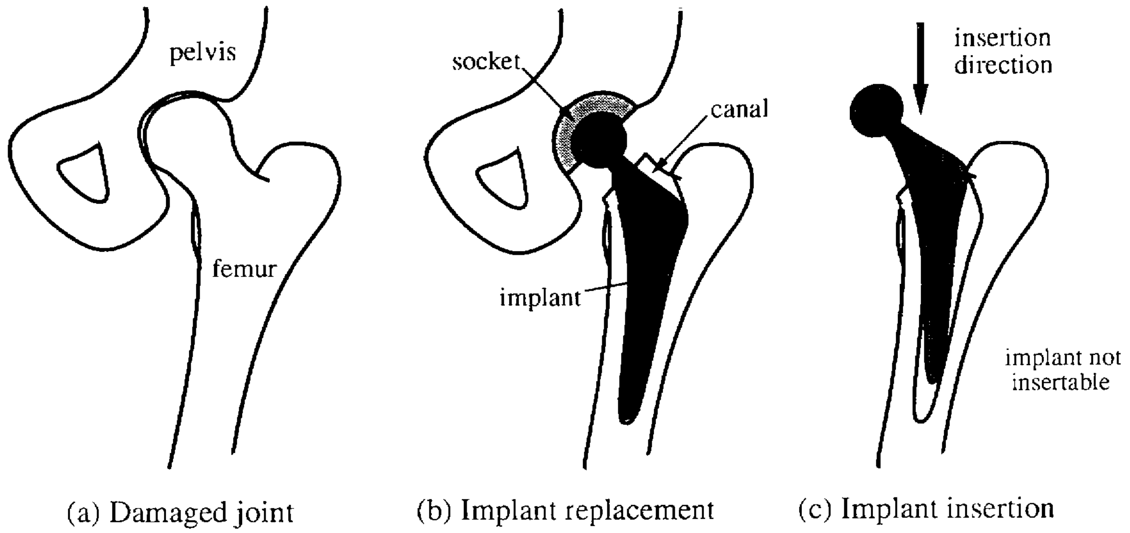

*Fig. 2. Illustration of a total hip replacement procedure. A damaged joint connecting the pelvis and the femur (*left*) is replaced by an artificial joint formed by a socket implanted in the pelvis and a ball mounted on a metal implant inserted into a canal carved in the thigh (*center*). Because of the desired tight fit between the implant and the canal, the insertion of the implant into the canal fails for certain shapes (*right).

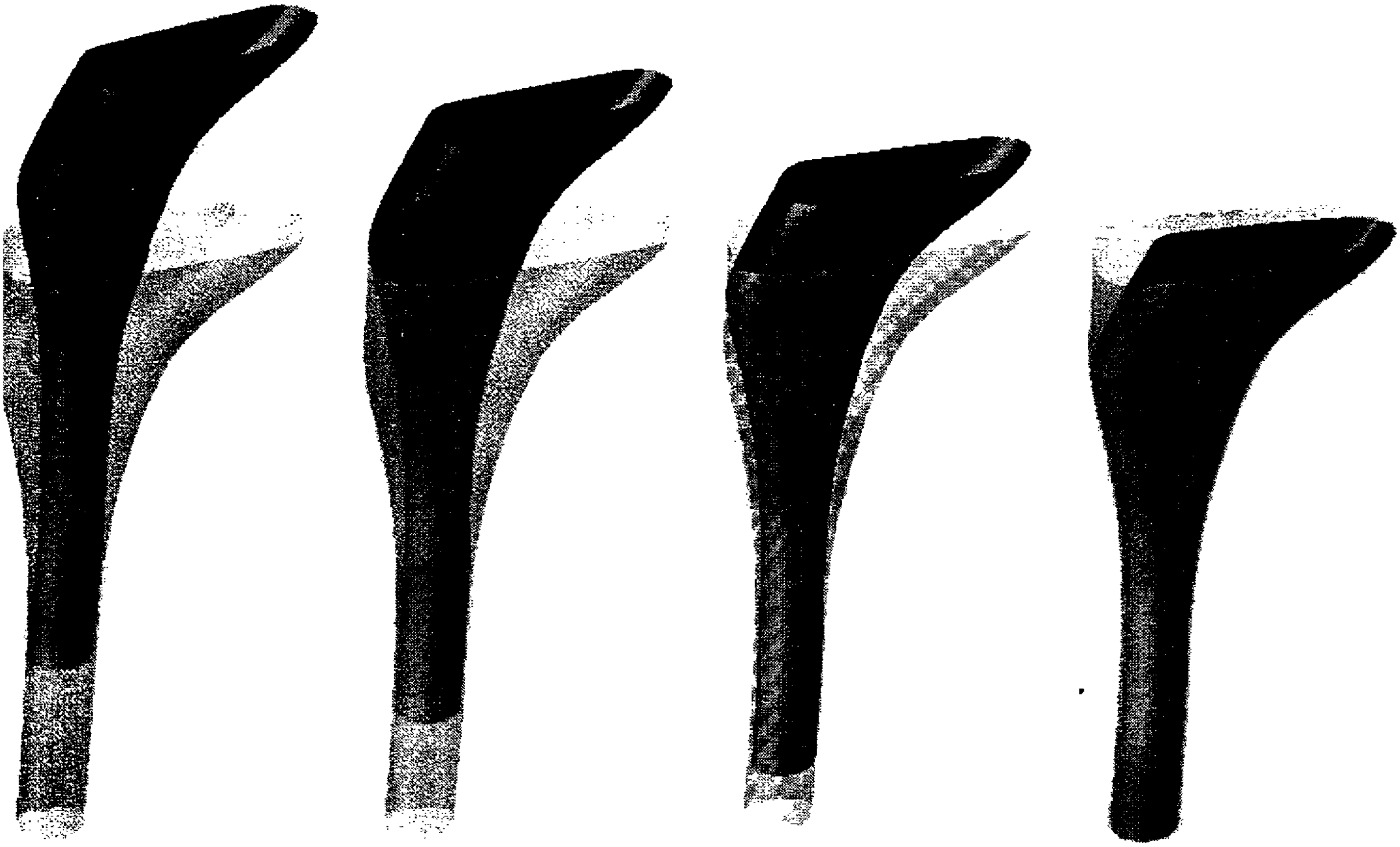

*Fig. 3. Snapshots of an insertion sequence of an implant stem (*dark body*) into a canal (*translucid body*) from the initial (*left*) to the final (*right*) configuration.*

as the search progresses. Global methods are by nature complete, whereas local methods are heuristic.

Global methods require computing and partitioning the configuration space, whose complexity is polynomial in the geometric size of the objects and exponential in their total number of degrees of freedom. Computing the entire configuration space is feasible only when its size is manageable, when most of it has to be searched to find a path, or when simplifications apply. It is impractical for tight-fit insertion problems with complex 3D shapes and six degrees of freedom because of the prohibitive number of configuration space cells and because only a fraction of them needs to be searched when a preferred insertion direction is known. Moreover, when the shapes are tightly fit and the insertion requires many small, incremental, coupled six-degree-of-freedom motions, the complexity of the configuration space cannot be reduced by approximating or abstracting the configuration space or by simplifying the shapes. Thus, techniques such as hierarchical configuration space decomposition (Brooks and Lozano-Perez 1983; Faverjon 1984), planning in low-dimensional configuration space projections (Buckley 1990), exploitation of the objects' geometric regularities (Giraud and Sidobre 1987), or randomized preprocessing of the configuration space (Kavraki and Latombe 1994) are not applicable.

Local strategies depend on the efficiency of the geometric computations and the effectiveness of the search strategy. Since the main difficulty of local search strategies is avoiding dead ends or local minima, existing local strategies emphasize search effectiveness. Donald's algorithm (Donald 1987) for a moving six-degree-of-freedom polyhedron creates a fine-resolution configuration space grid and uses heuristics based on the local configuration space geometry to search for a path through grid points. For tight fits, this method requires a high-resolution grid and, thus, very many small incremental motions to move even small distances. Potential field methods (Barraquand and Latombe 1991; Barraquand et al. 1992) place a potential field function in configuration space and search for an interference-free energy minimization path. They are applicable to multi-degree-of-freedom systems with moderate geometric complexity, as they require frequent object overlap tests or numerical potential field bitmap representations (Gupta and Zhu 1994). Since these methods rely on the minimization of a function that includes both the criterion describing the task (getting to the goal configuration) and the distance to the obstacles, whose task is to push moving objects away from them, they work only in relatively simple environments. In more complex environments this formulation leads to oscillations between opposite obstacle surfaces and undesired repulsion patterns, and prevents the moving object from getting arbitrarily close to the obstacles. These approaches are clearly impractical for problems with very complex body geometry and small clearances because they require many interference tests, many distance computations, or very high resolution bitmaps.

In this article we present a local path planning algorithm for the insertion problem that handles very complex body geometries. The algorithm uses limited, localized search, thereby exploiting the tight fit and known insertion direction. It efficiently performs local geometric computations to determine the series of small interference-free motion steps that are required for complex shapes and tight fit. Our approach is closest to that of Faverjon and Tournassoud (1987), which separates the noninterference constraints from the task constraints, thereby removing the undesired behaviors and providing better control. It translates the nonoverlap constraints into geometric constraints and factors them out of the function to minimize. The minimization function is formulated in terms of the task alone. In addition, our approach substitutes the expensive interference testing and distance computation of potential field methods with incremental maintenance of proximity relations that can be computed in worst-case time linearly proportional to the number of points in the body surface (Property 1, Section 4.2) and constant time in average. The algorithm contributes to research in path planning by providing:

- A practical solution to a well-defined problem: the insertion of a very complex three-dimensional body into a cavity with small clearances
- A configuration space cell decomposition based on neighborhood relations between body and cavity surface elements
- An efficient cell management method based on the locality principle
- A formulation of approximated localized linear configuration space constraints for small, coupled six-degree-of-freedom motions
- A formulation of linear programming problems to find the maximum allowed displacement in a preferred direction satisfying a set of motion constraints
- An incremental, greedy, path finding strategy with local backtracking that produces quasi-monotone insertion paths to any desired resolution

3. Solution Overview

We formulate the insertion problem as a path planning problem in configuration space. We find a path by incrementally constructing and searching configuration space cells, using the preferred insertion direction as a guide to construct cells and move within them. When no progress in the preferred direction is possible, we use limited, localized search to find alternative motions.

Configuration space cells are defined by proximity relations between body and cavity surface elements. Because of the tight fit between the body and the cavity, the motion of any point in the body surface is constrained by a cavity surface element in its immediate neighborhood. The pairing between body points and their closest cavity surface elements defines a neighborhood relation. A *neighborhood* is a subset of Euclidean space containing a single pair consisting of a body surface point and its closest cavity surface element. Small motions that keep each body point inside its neighborhood are constrained only by their closest cavity surface elements. The configuration space cell is the locus of body configurations that keep the body points in their cavity neighborhoods. The neighborhood relation has two important properties: it defines configuration space cells with a linear number of local configuration space constraints, and it provides a simple, local, cell adjacency criterion based on neighborhood adjacency in Cartesian space.

To facilitate path planning within a cell and between adjacent cells, we simplify cell geometry and topology by introducing shape and motion approximations. We approximate the body shape with points and the cavity shape with planar facets on their surfaces. We define neighborhoods with convex polyhedral volumes containing cavity facets. We linearly approximate small body motions. With these approximations, we define configuration space cells with local, linear configuration space constraints for small motions in the neighborhood of an interference-free configuration. Because the cells are defined by linear constraints, they are singly connected and convex. Thus, small motions between any two configuration space points within the cell are interference free to within the approximation.

We find an interference-free insertion path by computing a sequence of small, interference-free body motion steps. We compute each step by constructing the configuration space cell in the neighborhood of the current configuration and finding the maximum allowable displacement in the preferred motion direction. This displacement is computed by solving a linear optimization problem whose objective function is the preferred motion direction and the constraints are the cell's linearized configuration space constraints. When no progress in the preferred direction is possible, we search for alternative motions by modifying the insertion direction. This process is repeated with the new body configurations until either the final inserted body configuration is reached or until no further progress can be made.

The proposed method is resolution-sound but not complete. It will produce a guaranteed interference-free insertion path up to a prespecified resolution, but will find such a path only when limited local search and backtracking suffice. The algorithm is thus appropriate for situations in which insertion paths, when they exist, are quasi-monotone. (A path is *monotone* in a preferred direction if and only if it always shows progress along that direction.)

4. Problem Formulation and Properties

We begin by formulating the general path planning problem and show how this formulation is specialized to the problem of inserting a complex, tightly fit body into a cavity. Section 4.1 introduces our notation and presents the standard path planning problem formulation. Section 4.2 defines a new configuration space cell decomposition criterion based on proximity relations and neighborhoods. Section 4.3 introduces approximations that yield a cell decomposition satisfying two basic properties (Latombe 1991, Chapter 5): (1) the geometry of the cell is simple enough to make it easy to compute a path between two configurations in the cell, and (2) cell adjacency testing and path crossing of adjacent cell boundaries can be computed efficiently. Section 4.4 describes the path discretization criterion. Section 4.5 formulates the local search criterion as a function to be maximized, subject to cell constraints. This yields a formulation similar to the potential field method (Latombe 1991, Chapter 7).

4.1. General Formulation

We formulate the insertion problem as a motion planning problem in which the solid body B is a three-dimensional moving object with six degrees-of-freedom, and the cavity C is a fixed three-dimensional obstacle. We associate a body coordinate frame to the origin of the body and a cavity coordinate frame to the origin of the cavity. The cavity coordinate frame remains fixed, while the body coordinate frame moves with the body. The body and cavity shapes are described with respect to their coordinate frames. The position and orientation of the body, hereon referred to as the *configuration* of the body, is defined with respect to the cavity's fixed coordinate frame.

Let $(\boldsymbol{p},\boldsymbol{\theta})$ be the six configuration variables (three translations and three rotations) describing the configuration (position and orientation) of the body with respect to the cavity's fixed coordinate frame.[1] Let $\boldsymbol{F}(\boldsymbol{p},\boldsymbol{\theta})$ be the transformation mapping points in body coordinates to points in cavity coordinates in body position $\boldsymbol{p}$ and orientation $\boldsymbol{\theta}$. Let $\boldsymbol{b}$ be a body point whose coordinates are

[1] Notation conventions: bold lower case letters, $\boldsymbol{x}$, denote three-dimensional vectors. Bold lowercase letters with an overbar, $\bar{\boldsymbol{x}}$, denote n-dimensional vectors. $\mathbf{0}$ is an n-dimensional vector whose entries are all zero. Bold capital letters, $\bar{\mathbf{A}}$, denote matrices.

with respect to the body coordinate frame. The position $\boldsymbol{v}$ of a body point $\boldsymbol{b}$ in configuration $(\boldsymbol{p},\boldsymbol{\theta})$ with respect to the cavity's fixed frame is expressed as:

$$\boldsymbol{v} = F(\boldsymbol{p},\boldsymbol{\theta}) \cdot \boldsymbol{b} = Rot(\boldsymbol{\theta}) \cdot \boldsymbol{b} + \boldsymbol{p}, \qquad (1)$$

where $Rot(\boldsymbol{\theta})$ is the rotation operator specifying the orientation of the body with respect to the cavity's fixed coordinate frame.

Let B and C be the sets of three-dimensional points describing the body and the cavity. Let $H(\boldsymbol{x})$ be a function describing the shape of the cavity surface. A point $\boldsymbol{x}$ lies on or inside the cavity (and outside the cavity surface) when

$$H(\boldsymbol{x}) \leq 0.$$

A body point $\boldsymbol{b}$ in configuration $(\boldsymbol{p},\boldsymbol{\theta})$ lies on or inside the cavity when

$$H(F(\boldsymbol{p},\boldsymbol{\theta}) \cdot \boldsymbol{b}) \leq 0.$$

This condition, formulated over the set of all body points $\boldsymbol{b}$ on the surface of B, defines the body *configuration constraints*, which must hold for the body not to penetrate the cavity walls.

The set of body configurations for which the body and the cavity do not interpenetrate is defined as the set of positions and orientations for which all the points in the body surface lie on or outside the cavity surface. This set of configurations is called the *free configuration space*:

$$C_{\text{free}} = \{(\boldsymbol{p},\boldsymbol{\theta}) \mid H(F(\boldsymbol{p},\boldsymbol{\theta}) \cdot \boldsymbol{b}) \leq 0, \ \forall \boldsymbol{b} \in B\}.$$

The boundary of free configuration space, C_{contact}, is the set of configurations in which the body and the cavity are in contact.

We represent body motions as paths in configuration space. A path is a continuous function $T(t)$ specifying the position and orientation of the body at time t. It is interference-free if and only if the body does not penetrate the cavity at any time during the motion—that is, if all body configurations in the path are interference free. An interference-free path defines a mapping from the continuous unit interval to free configuration space:

$$T : [0,1] \rightarrow C_{\text{free}},$$

where $T(0) = (\boldsymbol{p}_0, \boldsymbol{\theta}_0)$ is the initial configuration at starting time 0 and $T(1) = (\boldsymbol{p}_f, \boldsymbol{\theta}_f)$ the final body configuration at ending time 1.

4.2. The Locality Principle

We now introduce the locality principle, which defines the partition criterion of the configuration space. It is based on the proximity relation between body and cavity surface elements. Because of the tight fit between the body and the cavity, the motion of any point in the body surface is constrained by a cavity surface element in its immediate neighborhood (Fig. 4). The pairing between body points in a given configuration and their closest cavity surface elements defines a neighborhood relation. A *neighborhood* is a subset of Euclidean space containing a single pair consisting of a body surface point and its closest cavity surface element. Small motions that keep each body point inside its neighborhood are only constrained by its closest cavity surface element.

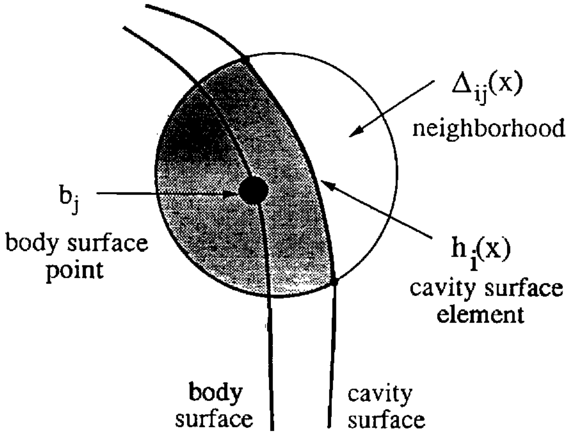

Fig. 4. Motions of the body point $\boldsymbol{b}_j$ in configuration $(\boldsymbol{p},\boldsymbol{\theta})$ are constrained by the closest cavity surface $h_i(\boldsymbol{x})$ in the neighborhood $\Delta_{ij}(\boldsymbol{x})$. The shaded area corresponds to the local configuration constraint $h_i(\boldsymbol{x}) \leq 0$.

It is important to note that our notion of neighborhood is based on the space surrounding *pairs of elements* (a body point and its closest cavity surface element), not just a single element, as is customary. For the purposes of the following discussion, neighborhoods can have any shape and can overlap, provided that only a single body surface point and a single cavity surface element are inside. For example, a neighborhood can be a sphere centered at body point $\boldsymbol{b}_j$ (see Fig. 4) and of radius larger than the distance to its closest cavity surface element $h_i(\boldsymbol{x})$. As we will see later, neighborhoods can be approximated (Section 4.3) and computed efficiently (Section 5.2). For each neighborhood, we formulate the local configuration space constraints of the body point and the cavity surface element in it. The conjunction of these constraints over all body points defines the set of legal body configurations for which there is no interference and the neighborhood relations are maintained.

Let $h_i(\boldsymbol{x})$ be a cavity surface element and $\boldsymbol{b}_j$ a body surface point in configuration $(\boldsymbol{p},\boldsymbol{\theta})$. Let the neighborhood $\Delta_{ij}(\boldsymbol{x})$ be a subset of Cartesian space containing *only* the pair consisting of the body surface point $\boldsymbol{b}_j$ in configuration $(\boldsymbol{p}, \boldsymbol{\theta})$ and its closest cavity surface element $h_i(\boldsymbol{x})$ (the neighborhood does not contain any other body surface points or cavity surface element). The *local configuration space constraint* imposed by the cavity surface element on the body point is defined by

$$h_i(F(\boldsymbol{p},\boldsymbol{\theta}) \cdot \boldsymbol{b}_j) \le 0 \tag{2}$$

for $F(\boldsymbol{p},\boldsymbol{\theta}) \cdot \boldsymbol{b}_j \in \Delta_{ij}(\boldsymbol{x})$.

A direct consequence of this definition is that the local configuration space constraint subsumes the configuration space constraints imposed by all other cavity surface elements in the neighborhood:

$$\begin{gathered} h_i(F(\boldsymbol{p},\boldsymbol{\theta}) \cdot \boldsymbol{b}_j) \le 0 \iff H(F(\boldsymbol{p},\boldsymbol{\theta}) \cdot \boldsymbol{b}_j) \le 0 \\ \forall(\boldsymbol{p},\boldsymbol{\theta}) \text{ such that } (F(\boldsymbol{p},\boldsymbol{\theta}) \cdot \boldsymbol{b}_j) \in \Delta_{ij}(\boldsymbol{x}). \end{gathered} \tag{3}$$

This is because a body point moving in the neighborhood will touch the cavity surface element defining the neighborhood and the constraint before any other surface element.

We define the configuration space cell as the locus of body configurations that keeps all body surface points inside their neighborhoods:

$$\begin{aligned} \mathcal{C} = \{(\boldsymbol{p},\boldsymbol{\theta}) \mid\ & h_i(F(\boldsymbol{p},\boldsymbol{\theta}) \cdot \boldsymbol{b}_j) \le 0, \\ & (F(\boldsymbol{p},\boldsymbol{\theta}) \cdot \boldsymbol{b}_j) \in \Delta_{ij}(\boldsymbol{x}), \quad \forall \boldsymbol{b}_j \in B,\ h_i(\boldsymbol{x}) \in H\} \end{aligned}$$

where H is a set of functions $h_i(\boldsymbol{x})$ describing the cavity surface. The neighborhood relations defining the configuration space cells provide, in Latombe's terminology (Latombe 1991), the *criticality condition* of the configuration space partition. They have two important properties:

PROPERTY 1. The configuration space cells are defined by local configuration space constraints. The number of constraints is linearly proportional to the number of body surface points.

This property follows directly from the definition of cell and from eq. (4).

PROPERTY 2. Two configuration cells can be adjacent only if all their neighborhoods are adjacent:

$$\begin{aligned} \text{adjacent}(\mathcal{C}_k, \mathcal{C}_l) \implies\ & \Delta^k_{ij}(\boldsymbol{x}) \cap \Delta^l_{ij}(\boldsymbol{x}) \neq \emptyset, \\ & \forall \boldsymbol{b}_j \in B,\ h_i(\boldsymbol{x}) \in H \end{aligned}$$

where $\Delta^k_{ij}(\boldsymbol{x})$ and $\Delta^l_{ij}(\boldsymbol{x})$ are the neighborhoods defining cells $\mathcal{C}_k$ and $\mathcal{C}_l$.

Proof. By definition of cell adjacency, two configuration space cells are adjacent if and only if there exists a direct path (a path not going through any other cell) between any configuration in one cell and any configuration in the other cell. By definition of cell, all body point configurations in the path belong to either one or the other cell. Because motions must be continuous, body configurations on the path continuously move body points from one neighborhood to the other. However, the motion cannot occur without discontinuities when two neighborhoods are not adjacent, as this involves going through a third configuration space cell. Thus, there is no direct path between the two original cells. □

These two properties have important computational advantages. The first defines configuration space cells with local configuration space constraints and requires only a number of configuration space constraints linearly proportional to the number of body points, rather than the product of the number of body points and cavity surface elements, as would be the case if all pairs had to be considered. The second provides a simple cell adjacency criterion based on neighborhood adjacency in Cartesian space. They both facilitate efficient incremental construction of adjacent configuration space cells.

4.3. Shape, Neighborhood, and Small Motions Approximations

To facilitate path planning within a cell and between adjacent cells, we simplify cell geometry and topology by introducing shape and motion approximations. These approximations yield a linear local configuration space constraint formulation for small motions.

We describe the shape of the body and the cavity by a finite set of surface elements to any desired resolution δ. We discretize the implant shape by sampling its surface with control points $\boldsymbol{b}_j$ such that:

$$\forall \boldsymbol{b}\ \ \exists \boldsymbol{b}_j \quad \|\boldsymbol{b}_j - \boldsymbol{b}\| \le \delta$$

We discretize the cavity shape H with a set of planar facets $h_i(\boldsymbol{x}) = \boldsymbol{a}_i \cdot \boldsymbol{x} - c_i$ such that the distance between the planar facet and the real surface does not exceed δ:

$$\forall \boldsymbol{x}\ \ \exists h_i(\boldsymbol{x}) \quad |\ H(\boldsymbol{x}) - \boldsymbol{a}_i \cdot \boldsymbol{x} - c_i)\ | \le \ \delta$$

The local configuration space constraint (eq. 2) becomes

$$\boldsymbol{a}_i \cdot (F(\boldsymbol{p},\boldsymbol{\theta}) \cdot \boldsymbol{b}_j) - c_i \le 0, \tag{4}$$

which is a linear expression in the configuration $(\boldsymbol{p}, \boldsymbol{\theta})$ of body point $\boldsymbol{b}_j$. It guarantees that the body point in the given configuration will not overlap the exact cavity surface element by more than δ.

To facilitate neighborhood construction and membership testing, we define neighborhoods as convex polyhedral volumes around cavity facets. The neighborhood is defined by n intersecting planar half spaces:

$$\mathbf{A}_i \cdot \boldsymbol{x} - \bar{\mathbf{c}}_i \le \mathbf{0}.$$

A body surface point $\boldsymbol{b}_j$ in configuration $(\boldsymbol{p}, \boldsymbol{\theta})$ is inside neighborhood $\Delta_{ij}(\boldsymbol{x})$ if and only if

$$\mathbf{A}_i \cdot (F(\boldsymbol{p}, \boldsymbol{\theta}) \cdot \boldsymbol{b}_j) - \bar{\mathbf{c}}_i \leq \mathbf{0}. \tag{5}$$

Note that when the cavity facet is one of the planes defining the neighborhood, the local configuration space constraint eq. (4), $\boldsymbol{a}_i \cdot (F(\boldsymbol{p}, \boldsymbol{\theta}) \cdot \boldsymbol{b}_j) - c_i \leq 0$ is one of the n inequalities defining the neighborhood. In this case, the set of inequalities in eq. (5) defines both the conditions for neighborhood membership and the local configuration space constraint. For simplicity, we will assume in the rest of this article that the cavity facet is one of the neighborhood boundaries.

Based on this observation, we can now define the approximation of the configuration space cell as

$$C' = \{(\boldsymbol{p}, \boldsymbol{\theta}) \mid \mathbf{A}_i \cdot (F(\boldsymbol{p}, \boldsymbol{\theta}) \cdot \boldsymbol{b}_j) - \bar{\mathbf{c}}_i \leq \mathbf{0}, \quad \forall \boldsymbol{b}_j \in B\}.$$

Note that the cell is defined by a set of linear inequalities in the configuration of body point $\boldsymbol{b}_j$ in configuration $(\boldsymbol{p}, \boldsymbol{\theta})$.

The transformation $F(\boldsymbol{p}, \boldsymbol{\theta})$ introduces a nonlinear term in the definition of the configuration cell, as it involves multiplying by a rotation matrix $Rot(\boldsymbol{\alpha})$ (eq. (1)). Given an interference-free configuration, we can approximate small motions from that configuration with a linear transformation and obtain a set of linear inequalities.

Let $\boldsymbol{v}_j$ be the position of point $\boldsymbol{b}_j$ in configuration $(\boldsymbol{p}_0, \boldsymbol{\theta}_0)$. The position of $\boldsymbol{v}_j^{'}$ of $\boldsymbol{b}_j$ after a small motion $(\boldsymbol{\epsilon}, \boldsymbol{\alpha})$ is given by:

$$\boldsymbol{v}_j^{'} = Rot(\boldsymbol{\alpha}) \cdot \boldsymbol{v}_j + \boldsymbol{\epsilon}$$

Since $(\boldsymbol{\epsilon}, \boldsymbol{\alpha})$ is a small motion, we can approximate it with the linear expression

$$\boldsymbol{v}_j^{'} \simeq (\boldsymbol{\alpha} \times \boldsymbol{v}_j) + \boldsymbol{v}_j + \boldsymbol{\epsilon}$$

for $\mid \boldsymbol{\epsilon} \mid \leq \epsilon_{\max}$ and $\mid \boldsymbol{\alpha} \mid \leq \alpha_{\max}$. Substituting this approximation into eq. (5), we obtain:

$$\begin{aligned} \mathbf{A}_i \cdot (F(\boldsymbol{p}, \boldsymbol{\theta}) \cdot \boldsymbol{b}_j) - \bar{\mathbf{c}}_i &\leq \mathbf{0}, \\ \mathbf{A}_i \cdot ((\boldsymbol{\alpha} \times \boldsymbol{v}_j) + \boldsymbol{v}_j + \boldsymbol{\epsilon}) - \bar{\mathbf{c}}_i &\leq \mathbf{0}, \\ (\boldsymbol{v}_j \times \mathbf{A}_i) \cdot \boldsymbol{\alpha} + \mathbf{A}_i \cdot \boldsymbol{\epsilon} - (\bar{\mathbf{c}}_i - \mathbf{A}_i \cdot \boldsymbol{v}_j) &\leq \mathbf{0}, \end{aligned} \tag{6}$$

where $\boldsymbol{v}_j = F(\boldsymbol{p}, \boldsymbol{\theta}) \cdot \boldsymbol{b}_j$. The result is a set of linear inequalities in the new motion parameters $(\boldsymbol{\epsilon}, \boldsymbol{\alpha})$, which correspond to the configuration space parameters:

$$F(\boldsymbol{p}, \boldsymbol{\theta}) = F(\boldsymbol{\epsilon}, \boldsymbol{\alpha}) \cdot F(\boldsymbol{p}_0, \boldsymbol{\theta}_0). \tag{7}$$

We can now obtain an approximation of the configuration space cell with a set of linear constraints. The approximated configuration space cell describes body configurations with an interference no greater than δ (plus the small error introduced by the small motion linearization) with the exact body and cavity cells.

PROPERTY 3. Configuration space cells can be approximated for small motions with a linear approximation of local configuration space constraints in the neighborhood of an interference-free configuration:

$$\begin{aligned} C' = \{(\boldsymbol{p}, \boldsymbol{\theta}) \mid (\boldsymbol{v}_j \times \mathbf{A}_i) \cdot \boldsymbol{\alpha} + \mathbf{A}_i \cdot \boldsymbol{\epsilon} \\ -(\bar{\mathbf{c}}_i - \mathbf{A}_i \cdot \boldsymbol{v}_j) \leq \mathbf{0}, \quad \forall \boldsymbol{b}_j \in B\}, \end{aligned}$$

where $\boldsymbol{v}_j = F(\boldsymbol{p}_0, \boldsymbol{\theta}_0) \cdot \boldsymbol{b}_j$ and $(\boldsymbol{p}_0, \boldsymbol{\theta}_0)$ is an interference-free configuration, $(\boldsymbol{p}, \boldsymbol{\theta})$ is obtained from $(\boldsymbol{\epsilon}, \boldsymbol{\alpha})$ by eq. (7), and $\mid \boldsymbol{\epsilon} \mid \leq \epsilon_{\max}$ and $\mid \boldsymbol{\alpha} \mid \leq \alpha_{\max}$. This property follows directly from eq. (6).

Property 3 has two important advantages. First, because the cells are defined by linear constraints, they are convex and singly connected. Thus, motions between any two configuration space points within the cell are guaranteed to be interference-free to within the approximation. In particular, the straight line (in configuration space) connecting any two configurations within a cell is interference-free. Second, it allows the incremental construction of motion paths by computing a sequence of small motions from interference-free configurations.

4.4. Path Discretization

We now introduce a discrete formulation of the insertion path, defining it in terms of a sequence of interference-free configurations belonging to adjacent configuration space cells.

An insertion path can be described to any desired resolution with a finite set of m interference-free configurations:

$$T(k) = (\boldsymbol{p}_k, \boldsymbol{\theta}_k)$$

such that

$$H(F(\boldsymbol{p}_k, \boldsymbol{\theta}_k) \cdot \boldsymbol{b}_j) \leq 0, \ \forall \boldsymbol{b}_j \in B$$

for $0 \leq k \leq 1$, where $T(0) = (\boldsymbol{p}_0, \boldsymbol{\theta}_0)$ and $T(1) = (\boldsymbol{p}_f, \boldsymbol{\theta}_f)$ are the initial and final body configurations.

The insertion path can be equivalently specified as a sequence of interference-free configurations belonging to adjacent configuration space cells defined by neighborhood relations. By Property 1, we can replace the configuration space constraint with local configuration space constraints defined in neighborhoods:

$$T(k) = (\boldsymbol{p}_k, \boldsymbol{\theta}_k)$$

such that

$$h_i(F(\boldsymbol{p}_k, \boldsymbol{\theta}_k) \cdot \boldsymbol{b}_j) \leq 0, (F(\boldsymbol{p}_k, \boldsymbol{\theta}_k) \cdot \boldsymbol{b}_j) \in \Delta_{ij}^k(\boldsymbol{x}), \ \ \forall \boldsymbol{b}_j \in B$$

for $0 \leq k \leq 1$, where $\Delta_{ij}^k(\boldsymbol{x})$ are the neighborhoods defined by cavity surface element $h_i(\boldsymbol{x})$ and body point

$\boldsymbol{b}_j$ in configuration $(\boldsymbol{p}_k, \boldsymbol{\theta}_k)$. By Property 2, the insertion path is guaranteed to be continuous when subsequent neighborhoods $\Delta_{ij}^k(\boldsymbol{x})$ and $\Delta_{ij}^{k+1}(\boldsymbol{x})$ are adjacent. This formulation yields the following important property:

PROPERTY 4. The problem of finding an insertion path to a desired resolution reduces to finding a discrete sequence of interference-free configurations in adjacent configuration space cells defined by neighborhood relations.

Using the shape, neighborhood, and small motion approximations described in the previous section, we can use a linear approximation to construct the configuration space cell and search for the next configuration in the path. By Property 3, the local configuration space constraints can be replaced by a linear approximation, yielding the insertion path approximation:

$$T(k) = (\boldsymbol{p}_k, \boldsymbol{\theta}_k)$$

such that

$$(\boldsymbol{v}_j^k \times \mathbf{A}_i^k) \cdot \alpha_k + \mathbf{A}_i^k \cdot \epsilon_k - (\bar{\mathbf{c}}_i^k - \mathbf{A}_i^k \cdot \boldsymbol{v}_j^k) \leq \mathbf{0},$$
$$\boldsymbol{v}_j^k = F(\boldsymbol{p}_k, \boldsymbol{\theta}_k) \cdot \boldsymbol{b}_j, \quad \forall \boldsymbol{b}_j \in B$$

where $(\boldsymbol{p}_k, \boldsymbol{\theta}_k)$ is obtained from (ϵ_k, α_k) by eq. (7), and $\mid \epsilon_k \mid \leq \epsilon_{\max}$ and $\mid \alpha_k \mid \leq \alpha_{\max}$.

This formulation leads naturally to an incremental path construction strategy. Starting from the initial configuration, we first construct the approximation of the configuration space cell containing it. We then compute a small motion by finding an interference-free configuration within the cell that satisfies a search objective, such as bringing the body closer to the final configuration by moving in a preferred direction. We can repeat this procedure with the new configuration, until a termination criterion—the final configuration is reached or no further progress is possible—is met.

Let $(\boldsymbol{p}_k, \boldsymbol{\theta}_k)$ be an interference-free configuration in the insertion path. The configuration $(\boldsymbol{p}_{k+1}, \boldsymbol{\theta}_{k+1})$ is determined by a small motion (ϵ_k, α_k) in the configuration space cell $\mathcal{C}_k$ corresponding to the neighborhood of $(\boldsymbol{p}_k, \boldsymbol{\theta}_k)$:

$$\mathcal{C}_k = \{(\boldsymbol{p}_{k+1}, \boldsymbol{\theta}_{k+1}) \mid \quad (\boldsymbol{v}_j^k \times \mathbf{A}_i^k) \cdot \alpha_k + \mathbf{A}_i^k \cdot \epsilon_k$$
$$-(\bar{\mathbf{c}}_i^k - \mathbf{A}_i^k \cdot \boldsymbol{v}_j^k) \leq \mathbf{0}, \ \ \forall \boldsymbol{b}_j \in B\}$$

where

$$\boldsymbol{v}_j^k = F(\boldsymbol{p}_k, \boldsymbol{\theta}_k) \cdot \boldsymbol{b}_j$$
$$(\boldsymbol{p}_{k+1}, \boldsymbol{\theta}_{k+1}) = F(\epsilon_k, \alpha_k) \cdot (\boldsymbol{p}_k, \boldsymbol{\theta}_k)$$

Because the configuration space cells are convex and simply connected, the straight line motion (in configuration space) from $(\boldsymbol{p}_k, \boldsymbol{\theta}_k)$ to any $(\boldsymbol{p}_{k+1}, \boldsymbol{\theta}_{k+1})$ is guaranteed to be interference free to within the resolution.

4.5. Moving in a Preferred Direction

We now specialize the path planning problem to the insertion task. Inserting a body into a cavity typically involves motion along a preferred direction. Although the precise path is not known, the main insertion direction is: it is usually defined by the major axis of the cavity and the body (i.e., the vertical axis in the examples of Fig. 1). Insertion paths follow this direction, with small, local corrections along the way. A good heuristic for local search is thus to move as far as possible along the preferred direction. Planning an interference free path with a preferred direction significantly reduces the search component of the planning algorithm. The preferred direction indicates which configuration space cells should be constructed and explored and how to move within each cell.

This heuristic strategy matches the incremental path construction method described in the previous section, which requires computing small motions in a preferred direction within a cell. We compute such motions by formulating and solving an optimization problem in which the objective function is the preferred motion direction and the constraints are the local configuration space constraints defining the cell. The solution to the optimization problem yields the maximum allowable displacement that satisfies the local configuration constraints and the neighborhood constraints. The new configuration, computed from the current configuration and the small motion, is guaranteed to be interference-free.

Let $(\boldsymbol{p}_k, \boldsymbol{\theta}_k)$ be the current interference-free body configuration. Let $\Delta_{ij}^k(\boldsymbol{x})$ be the neighborhoods defined for the $(\boldsymbol{p}_k, \boldsymbol{\theta}_k)$ body configuration. Let $\tau_k(\boldsymbol{p}, \boldsymbol{\theta})$ be the preferred direction function for step k. The furthest interference-free body configuration $(\boldsymbol{p}, \boldsymbol{\theta}) = (\boldsymbol{p}_{k+1}, \boldsymbol{\theta}_{k+1})$ in the preferred direction is obtained by solving the nonlinear optimization problem:

$$\textit{maximize } \tau_k(\boldsymbol{p}, \boldsymbol{\theta})$$
$$\text{subject to}$$
$$h_i(F(\boldsymbol{p}, \boldsymbol{\theta}) \cdot \boldsymbol{b}_j) \leq 0$$
$$(F(\boldsymbol{p}, \boldsymbol{\theta}) \cdot \boldsymbol{b}_j) \in \Delta_{ij}^k(\boldsymbol{x})$$

Substituting the local configuration space and neighborhood constraints with the approximations introduced in Section 4.3, we obtain the optimization problem LP_k:

$$\textit{maximize } \tau_k(\epsilon_k, \alpha_k) \qquad (8)$$
$$\text{subject to}$$
$$(\boldsymbol{v}_j^k \times \mathbf{A}_i^k) \cdot \alpha_k + \mathbf{A}_i^k \cdot \epsilon_k - (\bar{\mathbf{c}}_i^k - \mathbf{A}_i^k \cdot \boldsymbol{v}_j^k) \leq \boldsymbol{0}$$
$$\mid \epsilon_k \mid \ \leq \ \epsilon_{\max}$$
$$\mid \alpha_k \mid \ \leq \ \alpha_{\max}$$

When the objective function is linear, the problem is a linear optimization problem.

In this formulation, the objective function provides the local search criterion. For example, the function for moving in the preferred direction can be obtained from the vector difference of the initial and final configurations and the translational motion parameters:

$$\tau_k(\epsilon_k, \alpha_k) = \epsilon_k \cdot (\boldsymbol{p}_0 - \boldsymbol{p}_f) \qquad (9)$$

where $\boldsymbol{p}_0$ and $\boldsymbol{p}_f$ are the initial and final body positions. Changing the search criterion to allow for searching in other directions or for backtracking simply consists in changing the optimization function to reflect the new strategy.

Insertion paths produced by moving in a preferred direction are monotone: their subsequent body configurations show steady progress toward the final configuration by reducing the distance between the current body configuration and the final configuration. Quasi-monotone paths are obtained by occasionally varying the preferred direction. Monotone insertion paths have the advantage that they are easy to execute.

5. Insertion Algorithm

We now present a novel algorithm for computing an interference-free insertion path of a body into a cavity and discuss the algorithmic issues related to it. Section 5.1 describes the algorithm and characterizes its scope. Section 5.2 describes an efficient technique for neighborhood and cell management, a key subroutine of the insertion algorithm. Section 5.3 describes several search strategies to escape local minima.

5.1. Insertion Algorithm

Given a geometric description of the shapes of the body and the cavity, the initial and final configuration of the body, a desired shape resolution, and bounds on the maximum extent of small motion steps, the algorithm produces an interference-free path consisting of a sequence of interference-free configurations from the initial to the final configuration. The path is guaranteed to be interference free to within the resolution.

The algorithm is based on Property 4, which reduces the problem of finding an insertion path to a desired resolution to finding a discrete sequence of interference-free configurations in adjacent configuration space cells defined by neighborhood relations. Each new configuration is computed by formulating and solving an optimization problem. The algorithm starts with the initial body configuration and proceeds as follows:

1. Model the body and cavity shapes with body points and cavity surface elements.
2. Find, for each body point in the current configuration, its closest cavity surface and define their corresponding neighborhood.
3. Formulate an optimization problem with the local configuration and neighborhood constraints and the preferred direction of motion.
4. Compute a small motion step by solving the optimization problem.
5. Move the body by the small motion to the new configuration.
6. If the body has reached its final configuration, return the path.
7. Else if the body is stuck, modify the preferred direction of motion (if successive modifications have failed, return the path).
8. Return to Step 2.

The shape, neighborhood, and small motion approximations from Section 4.3 yield a linear formulation. The algorithm is described in Table 1. It formulates and solves a series of linear programming problems LP_k (eq. (9)) whose solution yields a small motion step that defines a new interference-free body configuration in the insertion path. The small motion step, which is a null step, the largest small step allowable, or an intermediate step, indicates how to proceed with the search and how to construct the next problem.

When the motion step is null, the body is stuck in the current configuration. At least one body point is in contact with a cavity facet, blocking the motion of the body in the preferred direction. No further motion is possible because there is no adjacent configuration space cell in the preferred motion direction. We can either abandon the search or attempt to move in another direction within the same configuration space cell. The new problem LP_{k+1} is formulated with the same constraints as LP_k, but with a new objective function τ_{k+1}. When the motion step is the largest small step allowable, the insertion path can proceed within the current configuration space cell. The new problem LP_{k+1} is formulated with the same neighborhoods $\Delta_{ij}^k(\boldsymbol{x})$, recomputing the local configuration and neighborhood constraints approximations in the new configuration with the new body point positions, $\boldsymbol{v}_j^{k+1} = F(\epsilon_k, \alpha_k) \cdot \boldsymbol{v}_j^k$. Finally, when the body motion is neither null nor the largest, at least one body point has reached the boundary of its neighborhood. Any further motion of the body in that direction will take that point outside the neighborhood, thereby violating the neighborhood constraints. A transition to the adjacent configuration space cell is necessary to continue the insertion path. The new problem LP_{k+1} is formulated by finding new neighborhoods $\Delta_{ij}^{k+1}(\boldsymbol{x})$ adjacent to the current neighborhoods $\Delta_{ij}^k(\boldsymbol{x})$ in the preferred direction of motion and formulating the corresponding local config-

Table 1. Incremental Insertion Algorithm.

```
1. Approximate the body surface with body points; approximate the cavity surface with planar cavity facets
2. Set $(\boldsymbol{p}_0, \boldsymbol{\theta}_0)$ to the initial configuration and $k$ to 0
3. Find initial neighborhoods and formulate $LP_0$
4. While body has not reached final configuration do
    (a) Solve $LP_k$, to obtain the small motion step $(\epsilon_k, \alpha_k)$
    (b) if the motion step is null
        (1) Declare the body struck and return fail, or
        (2) Formulate $LP_{k+1}$ with the constraints of $LP_k$ and a new objective function
    (c) if the motion step is the maximum allowable,
        formulate $LP_{k+1}$ with current neighborhoods $\Delta_{ij}^k(\boldsymbol{x})$
        and new body configurations $\boldsymbol{v}_j^{k+1} = F(\epsilon_k, \alpha_k) \cdot \boldsymbol{v}_j^k$
    (d) else
        formulate $LP_{k+1}$ with new neighborhoods $\Delta_{ij}^{k+1}(\boldsymbol{x})$ adjacent to $\Delta_{ij}^k(\boldsymbol{x})$
        and new body configurations $\boldsymbol{v}_j^{k+1} = F(\epsilon_k, \alpha_k) \cdot \boldsymbol{v}_j^k$
    (e) Compute the new configuration $(\boldsymbol{p}_{k+1}, \boldsymbol{\theta}_{k+1})$ and add it to the path
    (f) Increment $k$ by one
5. Return insertion path $(\boldsymbol{p}_k, \boldsymbol{\theta}_k)$
```

uration and neighborhood constraints for the new body configuration.

The solution of problem LP_k determines which scenario occurs and provides the information to formulate LP_{k+1}. The basis of linear programming problem LP_k indicates which inequality constraints are active (the equality condition holds). Thus, if one or more of the small displacement constraints is active, the implant has moved by the maximum displacement step. If one or more local configuration constraints are active, the corresponding implant control points have reached their neighborhood boundary. Zero motion parameter values indicate that no motion in the preferred direction is possible.

Properties 2, 3, and 4 guarantee that the insertion path computed by the algorithm is continuous and interference free to within the desired resolution. Property 1 guarantees that the size of the linear programming problems LP_k is linearly proportional to the number of body points. The number of body configurations in the insertion path, and thus the complexity of the algorithm, depends on the distance between the initial and final configuration, on the resolution of the body and cavity shape approximations, on the size of the neighborhoods, and on the amount of backtracking necessary. The Appendix formally defines a tight-fit measure, and the ϵ-approximation of path configuration spaces and establishes the relation between the configuration space complexity and the tight-fit measure.

The algorithm is resolution sound but not complete: it will produce a guaranteed interference-free insertion path up to a prespecified resolution, but will find such a path only when limited local search and backtracking suffice. The algorithm is thus appropriate for situations in which insertion paths, when they exist, are quasi-monotone. When the object shape approximations are conservative (i.e., the exact shape of the cavity is a subset of the approximated cavity shape and the approximate shape of the body is a subset of the exact cavity shape), the configuration space approximation is also conservative: it is a subset of the exact configuration space. Thus, if an approximated insertion path is found, it is guaranteed to be interference-free for the exact shapes. On the other hand, failure to find an approximated insertion path at a given resolution does not mean that one does not exist at a higher resolution.

5.2. Neighborhood and Cell Management

Key to the efficiency of the algorithm is the incremental creation and management of cavity neighborhoods and the configuration space cells they define. The algorithm must keep track, for every body configuration, of the closest cavity surface of each body point and its corresponding neighborhood. It must find neighborhoods adjacent to current neighborhoods in a preferred direction of motion. Neighborhoods can be managed statically or dynamically.

Static management precomputes fixed cavity neighborhoods. Before path computation, it uniformly partitions the cavity volume by associating a fixed neighborhood to each cavity element approximated to the prespecified resolution and recording the adjacencies between neighborhoods. During path computation, body points are associated with the neighborhoods they are in. The body points, their neighborhoods, and the body configuration define the cell constraints. Because body points can only move to adjacent neighborhoods at each step, the mapping between body points and neighborhoods is updated in constant time for each body point. Only the constraints associated with body points that migrated to new neighborhoods need to be changed to define the new optimization problem LP_{k+1}.

Figure 5 shows an example of cavity partition into pie slice–shaped volumes. The neighborhoods are constructed by creating a regular point mesh on the surface of the cavity to the desired resolution. Four adjacent points on the mesh (two up and two down) define a planar cavity facet. The cavity facet, together with four additional facets constructed with two additional points on the axis of the cavity, define a neighborhood described by a 3×5 matrix $\mathbf{A}_i$ and a five-dimensional vector $\bar{\mathbf{c}}_i$ such that $(\mathbf{A}_i \cdot \boldsymbol{x} - \bar{\mathbf{c}}_i \leq \mathbf{0}, \quad 1 \leq i \leq 5)$. Neighborhood adjacencies—up, down, left, and right—are directly determined from the mesh.

Static management avoids finding and constructing neighborhoods for all body points at each step. However, it has two disadvantages. First, because it precomputes neighborhoods regardless of the distance of the body point to the cavity surface, it approximates the cavity shape for the worst case, sometimes at a resolution much smaller than required. Small neighborhoods shorten the extent of small motion steps, thus requiring more motion steps to move from the initial to the final configuration. Second, when many body points are close to their neighborhood boundaries, many neighborhood migrations might be necessary before a significant small step can be taken. One way of overcoming this drawback is to have neighborhoods overlap, so that many neighborhood migrations happen at once for a single motion step.

Dynamic management overcomes these drawbacks by computing neighborhoods anew at each step. Given a body configuration, it determines the extent of each body point's neighborhood from its nearest cavity walls, its distance from them, and the required resolution. This optimizes the size of the neighborhood—and thus the size of the motion step—at the expense of geometric computations to determine neighborhood extensions and adjacencies.

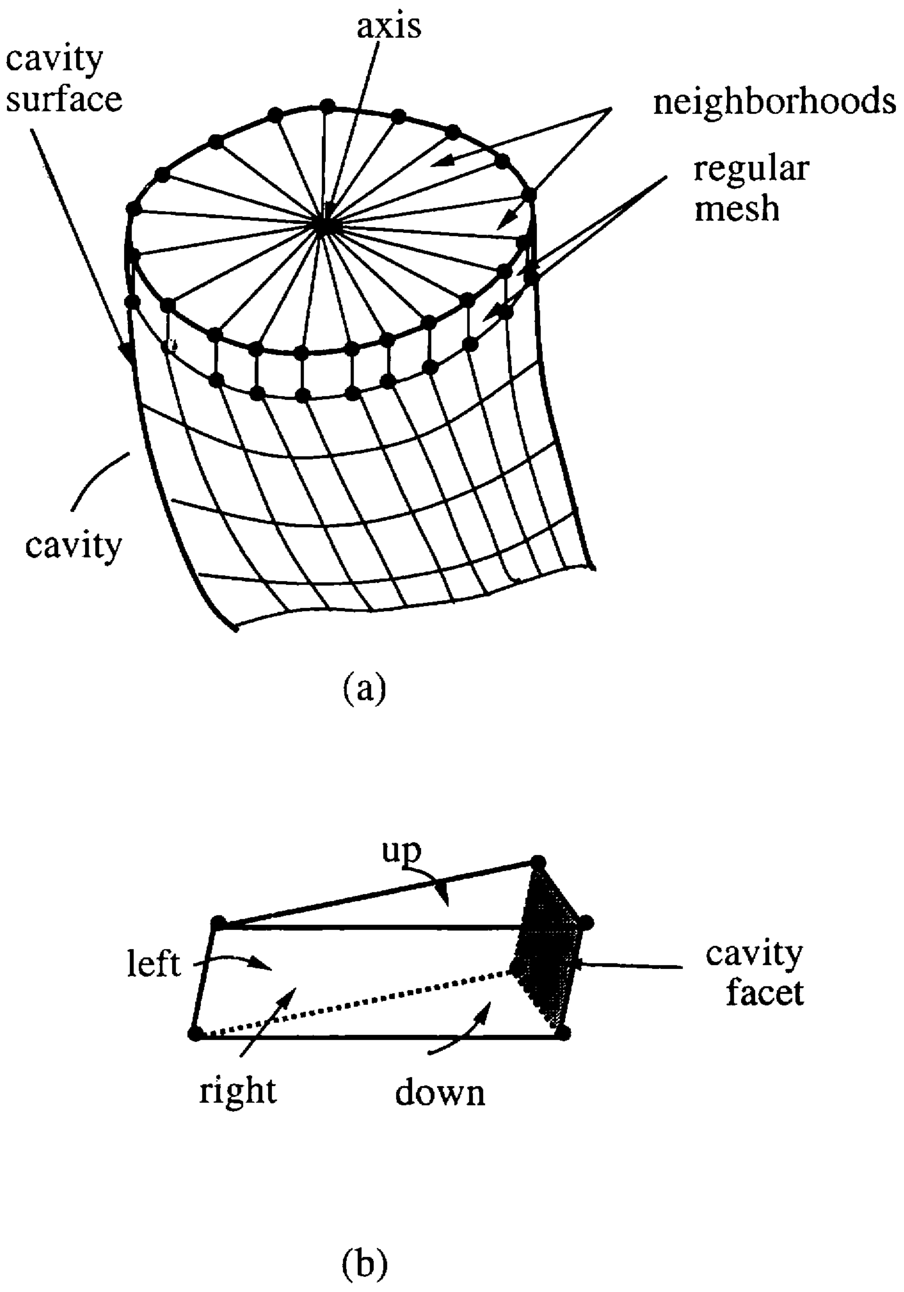

Fig. 5. Cavity partition into fixed neighborhoods. (A) Partition along the cavity center axis and surface. (B) Planes defining the neighborhood.

5.3. *Search Strategies*

An effective strategy in solving the insertion problem is to start from the final, inserted configuration and attempt to extract the body from the cavity until the body is completely outside the cavity. Although reaching the exact inserted configuration is important, any starting initial position above the cavity is acceptable. It is thus simpler to start from the exact inserted configuration and move in the preferred direction until, for example, a prespecified height is reached. The insertion path is directly obtained from the extraction path by reversing the extraction path. The insertion motion is simply the extraction backward in time.

When moving in the preferred direction is no longer possible, the search strategy must be temporarily changed by modifying the objective function $\tau_k(\epsilon_k, \alpha_k)$ for one or more steps. Various path search strategies can be approximated by setting the weights of the six motion parameters differently. For example, the default objective function (eq. (9)) favors translations along the preferred direction while leaving open the choice of angles by setting the angle coefficients to zero. Reversing the sign of the motion parameter weights implements backtracking. Strategies are combined by combining weights. For example, a "wobbling" effect can be added by varying the weights of the rotation parameters according to a sine function:

$$\begin{aligned}\tau_k(\boldsymbol{\epsilon}_k, \boldsymbol{\alpha}_k) = \boldsymbol{\epsilon}_k \cdot (\boldsymbol{p}_0 - \boldsymbol{p}_m) + \boldsymbol{\alpha}_k \\ \cdot(p * sin(c\psi_0), p * cos(c\psi_0), 1),\end{aligned}$$

where p and ψ_0 are the pitch and angular increment of the screw motion and c is a constant. Similarly, a "repulsive force" effect can be obtained by adding the inverse weighted sum of the distance between the point and the wall and the vector normal to the plane. Path search strategies are problem dependent and are thus best adjusted for particular tasks and situations.

6. Insertability Analysis of Custom Hip Implants

We have implemented the insertion algorithm and demonstrated its use in analyzing the insertability of cementless custom hip implants (Section (1)). The purpose of insertability analysis is to determine whether an implant can be inserted without interferences into a canal carved in the bone (see Fig. 2). Verifying implant insertability helps validate shape designs, identify wedging configurations and interfering surfaces, and support shape modification and redesign.

About half of the approximately 300,000 total hip replacement (THR) surgeries performed each year use cementless implants, in which the stem of the implant fits tightly into a matching canal carved in the shaft of the femur. In an increasing number of cases, custom, patient-matched implants are designed for each patient from computed tomography (CT) data (Stulberg 1989). The goal of the design is to obtain the tightest possible fit in the implant's final (working) configuration inside the canal. Tight fit provides mechanical stability and adequate stress distribution transfer; promotes bone ingrowth onto the implant; and avoids cement, which occasionally provides poor fixation and deteriorates over time. To obtain the desired fit and avoid splitting the femur during insertion, the implant must be insertable without interferences to its final working position. These requirements often lead to unique, very complex body and canal shapes with very small clearances.

Insertability analysis is an important and difficult aspect of custom implant design. Small clearances (less than 0.01 inch), complex shapes, and high accuracy in implant machining and robot-assisted bone canal preparation (Paul et al. 1992; Taylor et al. 1994) greatly complicate the task. Manual shape design validation and wedging configuration and interfering surfaces identification (as illustrated in Figures 1c and 3) are impractical. Simple solutions, such as computing a swept volume for a straight line insertion path, are either inapplicable or pose significant constraints on implant design.

The incremental insertion algorithm is especially suited for preoperative implant insertability analysis. The quasistatic rigid body geometric model is appropriate because the implant is made of metal, a significant part of the canal is hard bone, and the insertion motion is slow (quasi-static). The implant and canal three-dimensional shapes are complex, requiring about 10,000 points and facets to obtain accuracy of 0.01 inch. The fit is tight, with a clearance of about 0.01 inch. The preferred insertion direction is clearly defined, but because of the tight fit, small, coupled, six-degree-of-freedom motions are necessary to insert the implant into the canal. The insertion path should be quasi-monotone and consist of small, incremental motions to avoid complex, time-consuming maneuvers and excessive manipulation and search by the surgeon during manual insertion. Computing an extraction path is simpler, as any starting implant configuration above a prespecified height is acceptable.

6.1. *EXTRACT: Program Characteristics*

The program, called EXTRACT, outputs an interference-free extraction path to the desired resolution. When the implant is not extractable, it stops at the stuck configuration and identifies the implant and canal surfaces causing the interference. It shows a graphical animation of the extraction and computes path statistics. EXTRACT is written in C and uses the IBM's Optimization Subroutine Library (OSL) to solve the linear optimization programs and the Graphics Library (GL) to show color 3D animation of the extraction.

EXTRACT computes quasi-monotone extraction paths using static, overlapping neighborhoods (Section 5.2). It inputs the implant and canal surface shapes designed on a CAD workstation from CT data and the desired shape approximation resolutions. The shapes are represented either as regular surface point meshes or as stacks of two-dimensional parallel contour slices defined by cubic splines forming a close contour. The initial configuration has the implant fit into the canal in its working position. The program computes first the body points and the pie slice–shaped canal neighborhoods from the CAD data. It optimizes body point spacing and neighborhood size, creating the largest elements with the smallest deviation between them. It computes the pairing between body points and cavity neighborhoods for the initial (inserted) configuration and maintains a data structure that records the correspondences as the body moves using the neighborhood adjacency relations.

Each linear problem LP_k is formulated from the current body configuration and the body point and cavity neighborhood constraints, adding to it the small motion constraints. Because each neighborhood is defined by five planes, LP_k has $5n + 12$ constraints in six variables, where n is the number of body points. Because the problem is highly redundant, we solve the dual problem and map the results back to the primal formulation. The program allows for slight implant and cavity overlap due to design and approximation errors by relaxing the local configuration constraints by a user-specified small positive constant c. This amounts to replacing the inequalities:

$$h_i(F(\boldsymbol{p},\boldsymbol{\theta}) \cdot \boldsymbol{b}_j) \leq 0$$

with

$$h_i(F(\boldsymbol{p},\boldsymbol{\theta}) \cdot \boldsymbol{b}_j) \leq c$$

for c > 0. This overlap allowance is also used to model bone compliance.

The program uses a simple resolution-sensitive search strategy, which adds a "wobble" component to the preferred direction of motion to avoid local maxima. When no further progress is possible, the program switches to the repulsive forces strategy for several steps (determined by the resolution) to momentarily move the implant away from the cavity surface and find a more favorable configuration to continue the extraction. Progress along the preferred direction is monitored over step intervals to decide when to terminate the search.

6.2. Experiments and Evaluation

EXTRACT has been successfully tested on 34 data sets at several resolutions. Four are synthetic examples, such as the screw and the boot in Fig. 1. Thirty are real hip implant designs provided by a medical equipment manufacturer. The synthetic examples were designed to either have known insertion paths (the screw) or to get stuck at known configurations. No insertability information was provided for the implant and canal data sets.

The typical implant and canal height is 4 inches, with cross-section diameters varying from 0.5 to 2 inches. Their shapes are described with 50 to 100 two-dimensional contours defined from CT slices spaced 0.025 to 0.1 inches apart. Each contour is described with 25 to 100 splines forming a closed contour. The canal and implant shapes are almost complementary, except for the upper part of the stem (Figure 1c and Figure 3, rightmost image). The implant and canal have no clearance in the final inserted configuration (some cases even have a small interference of up to 0.01 inch). The clearance is extremely small throughout most of the extraction path, ranging between 0 and 0.02 inch. The critical implant configurations near the inserted configuration (Figure 3, second from rightmost image). At a height of about 2.5 inches from the bottom of the canal, the implants can be directly pulled out of the canal with a straight vertical motion (Figure 3, leftmost image).

All data sets were tested at several resolutions ranging from 0.005 to 0.05 inch, with overlap allowances ranging from 0.01 to 0.05 inch. Resolutions of 0.03 inch were considered sufficient by the medical equipment manufacturer. The resolutions required implant and canal models with 1,000 to 10,000 implant points and cavity facets. The facet width, which determined the width of the pie slice–shaped cavity neighborhoods, ranged from 0.02 to 0.6 inch (average 0.1 inch), for a resolution of 0.05 inch to 0.003 to 0.2 inch (average, 0.025 inch), for a resolution of 0.005 inch. The facet height, determined by the spacing between slices, ranged from 0.025 inch to 0.1 inch for all resolutions.

The small motion limits were set to 0.1 inch for translations and one degree for rotations. The program stopped when the implant configuration reached a height of 2.5 inches from the bottom of the canal. Between 50 and 150 steps were required for resolutions of 0.05 inch. Higher resolutions (0.01 inch) required between 300 and 2,000 steps to either find a path or declare the implant stuck. Running times ranged from 3 to 45 minutes on an IBM RS/6000 Model 530 workstation with 64 MB of main memory, with most of the time spent on path computation.

In all but two cases, the implant extraction paths were successful. The successful cases were validated visually and quantitatively by measuring the maximum amount of overlap in the path configurations. In no case did the amount of overlap exceed the specified overlap allowance. Only a couple of paths were monotone; all the rest required minor backtracking. For the stuck cases, we verified that no extraction was possible by manually exploring alternate preferred motion directions, by manually choosing possible configurations and measuring overlap, and by restarting the search at different interference-free intermediate configurations. None of these strategies yielded a successful extraction path (Fig. 6). These results were confirmed by expert implant designers. The four synthetic cases yielded the expected results at all resolutions. Overall, these results provide evidence that the limited search strategy proved effective for practical implant insertability analysis.

Table 2 shows a sample of runs on several data sets. Note that the model size grows roughly linearly with the resolution, but the running time and the number of steps grows faster. This is because the neighborhood sizes shrink, so more cell changes are necessary to cover the same distance. Furthermore, the complexity of solving each linear programming problem LP_k grows roughly quadratically with the number of implant points. Because there was frequently no clearance (even some overlap in most path configurations), the value of the maximum overlap allowance (third column in the table) is greater than zero. When the actual clearance between the implant and the canal is smaller than the implant and canal resolution, the approximation can lead to false-negative results (implant 3/canal 3). However, the results are always reliable when its value exceeds the amount of initial overlap.

7. Conclusion and Extensions

We have presented a novel path planning algorithm for computing an interference-free insertion path of a body into a cavity to any desired resolution. The algorithm computes a sequence of interference-free configurations by incrementally constructing and searching configuration space cells defined by proximity relations between body points and cavity facets. It uses a predefined preferred insertion direction as a guide to construct adjacent cells and move within them. When no progress in the preferred direction is possible, limited, localized search is used to find alternative motions.

The algorithm contributes to research in path planning by providing a method for practically handling the geometric complexity of tight-fit insertions. Unlike most existing path planning methods, the algorithm is designed to handle very complex three-dimensional bodies requiring fine, complex, coupled six-degree-of-freedom motions in a preferred direction. It emphasizes local geometry and motion constraint computation over search. Tight fits require shape approximations consisting of thousands of facets to obtain reliable results. Their configuration space consists of narrow channels that get blocked with small shape variations or approximation errors. Moving through the channels amounts to following its walls (the configuration space constraints) in the preferred direction.

The algorithm uses shape and small motions approximations to define configuration space cells with local, linear configuration space constraints for small motions in the neighborhood of an interference-free configuration. By keeping track of proximity relations between body and surface elements, it identifies redundancies and reduces the number of local configuration space constraints in a cell from quadratic to linear (in the number of body points). Small motions in a preferred direction are computed by formulating and solving a linear optimization problem whose objective function is the preferred direction and whose constraints are the cell constraints.

We contemplate several extensions to the insertability analysis problem. To model slight body compression and deformation, we allow a small amount of interpenetration between body points and cavity facets by relaxing the local configuration constraints with a small user-defined positive constant. To incorporate user-defined path and configuration constraints, such as ranges of allowable body positions and orientations or bounding variations between consecutive configurations, we formulate the corresponding linear constraints and add them to each LP_k problem. Both extensions are practically useful and are readily incorporated into the algorithm. Dynamics modeling and stress analysis require modeling the forces between surface elements in contact to derive the stress distribution and the resultant insertion force. The insertion algorithm is useful to identify the surface elements in contact at each configuration from solving each LP_k problem and to provide the geometric information to formulate the dynamics problem and add friction (Erdmann 1994).

The insertion algorithm can serve as the basis for many related insertability analysis tasks, including design validation, tolerancing, and shape modification and optimization. Design validation is performed by computing an insertion path to within a prespecified resolution. Tolerancing is performed by testing the insertability of small variations of nominal body and cavity shapes. Shape modification and optimization are performed by identifying the stuck configuration and the surfaces causing the interference and then locally modifying the shapes around these surfaces.

8. Acknowledgments

We thank Troy Hershberger from Biomet Inc. for providing expertise on custom hip implants, the real data sets, and feedback on EXTRACT. We thank Gabriel Taubin for discussions and advice on all aspects of this work.

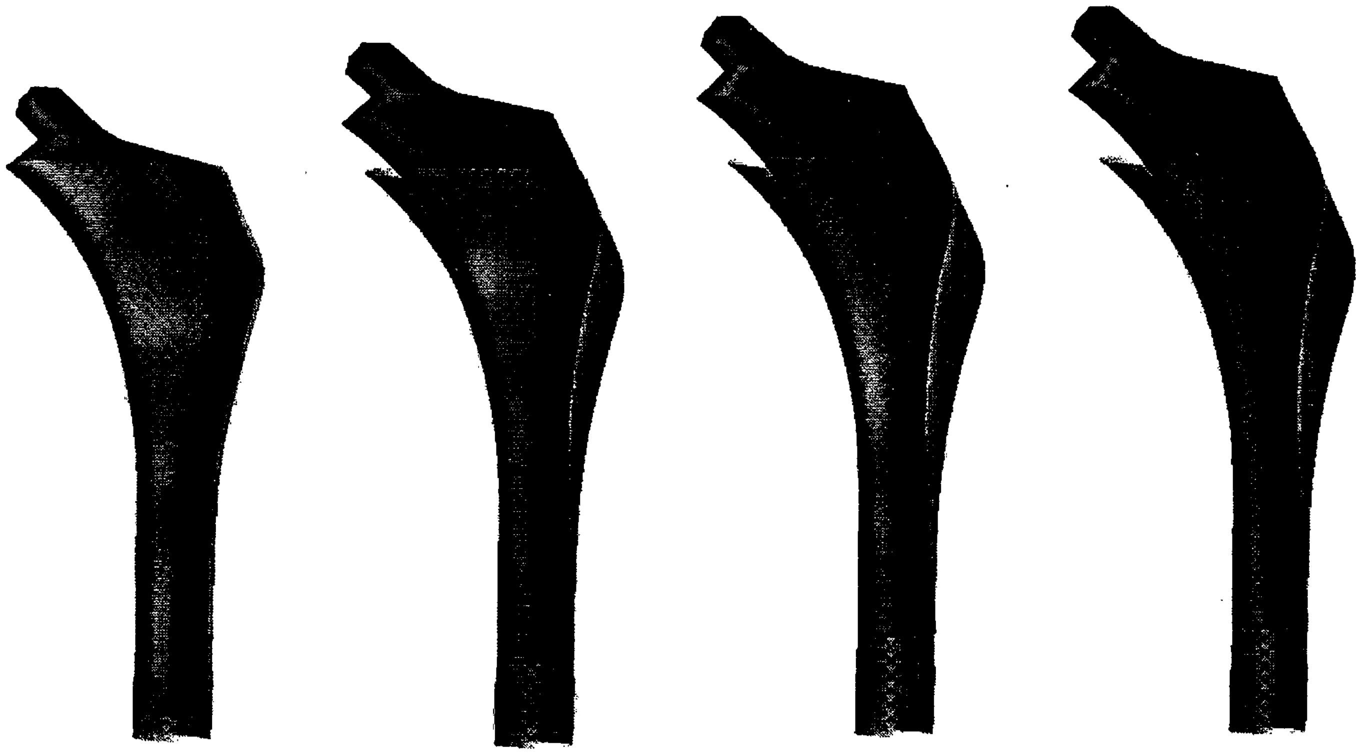

*Fig. 6. Snapshots of an extraction sequence of an uninsertable implant stem (*dark body*) into a canal (*translucid body*) from the initial (*left*) to the final (*right*) configuration. Local search could not find an extraction sequence.*

Table 2. Sample Test Cases.*

Case	Resolution	Overlap	Size	Steps	Time	Extraction
Boot-in/boot-out	0.025	0.01	1122	136	2:18	Yes
	0.01	0.0	1683	145	4:08	Yes
	0.005	0.0	3162	215	6:41	Yes
Implant 2/canal 2	0.05	0.02	1400	81	2:58	Yes
	0.025	0.015	2296	138	3:47	Yes
	0.01	0.015	3808	602	26:04	Yes
Implant 3/canal 3	0.05	0.02	1958	38	1:10	No (0.57)
	0.025	0.02	2848	52	3:07	No (0.67)
	0.01	0.02	5251	826	49:13	Yes
Implant 4/Canal 4	0.05	0.015	1840	116	3:20	Yes
	0.025	0.015	2640	165	8:20	Yes
	0.01	0.015	4720	698	38:31	Yes
Implant 5/Canal 5	0.05	0.05	2184	192	3:22	No (1.31)
	0.025	0.05	3367	187	6:35	No (0.76)
	0.01	0.05	6552	562	24:11	No (0.36)

*The columns list the test case, the implant and canal resolution in inches, the maximum overlap allowance, the number of implant points and cavity facets, the number of steps in the path, the execution time in minutes, and whether the implant was successfully extracted. The numbers in parentheses indicate how far the implant got before being stuck.

Appendix: Tight Fit and ϵ-Approximation

This appendix formally defines a tight-fit measure and the approximation of configuration space and establishes the relation between the approximated configuration space complexity and the tight-fit measure. Let T be an interference-free path between the initial body configuration $T(0) = \overline{q}_0$ and the final body configuration $T(1) = \overline{q}_f$. We define the clearance of a path configuration $T(t)$ as the Euclidean distance in configuration space from the path configuration to the closest point on the boundary of free space:[2]

$$\mathit{clearance}(T(t)) = \min_{\overline{q} \in \mathcal{C}_{\text{contact}}} \|T(t) - \overline{q}\|.$$

We define the path clearance as the smallest path configuration clearance:

$$\mathit{path\text{-}clearance}(T, \overline{q}_0, \overline{q}_f) = \min_{t \in [0,1]} \mathit{clearance}(T(t)).$$

We define a fit measure between two configurations as the largest path clearance over all interference-free paths connecting them:

$$\mathit{fit\text{-}measure}(\overline{q}_0, \overline{q}_f, \mathcal{C}_{\text{free}}) = \max_{T \text{ in } \mathcal{C}_{\text{free}}} \mathit{path\text{-}clearance}(T, \overline{q}_0, \overline{q}_f).$$

We can easily extend this definition to sets of initial and fin al configurations and to subsets of free configuration space:

$$\mathit{fit\text{-}measure}(R_0, R_f, \mathcal{C}) = \max_{T \text{ in } \mathcal{C}} \mathit{path\text{-}clearance}(T, \overline{q}_0, \overline{q}_f)$$

where $\overline{q}_0 \in R_0$ and $\overline{q}_f \in R_f, R_0, R_f \subseteq \mathcal{C} \subseteq \mathcal{C}_{\text{free}}$

We say that a body and a cavity fit have a tight-fit insertion path when the fit measure of their configuration space is much smaller than any of the body or cavity dimensions (their height, width, or length), or is much smaller than the distance between the initial and the final configuration:

$$\mathit{fit\text{-}measure}(\overline{q}_0, \overline{q}_f, \mathcal{C}_{\text{free}}) \ll \mathit{dimensions}(B) \ll \|\overline{q}_0 - \overline{q}_f\|$$

To ensure that insertion paths can be computed approximately, the configuration space approximation must be topologically equivalent and within an ϵ distance of the exact configuration space. This guarantees that no configuration space "channel" is blocked by the approximation.

Let $\mathcal{C}_{\text{approx}}$ be an approximation of $\mathcal{C}_{\text{free}}$. We say that $\mathcal{C}_{\text{approx}}$ is an ϵ-approximation of $\mathcal{C}_{\text{free}}$ for paths from an initial configuration $\overline{q}_0$ to a final configuration $\overline{q}_f$ if and only if for every interference-free path $T(t) \in \mathcal{C}_{\text{free}}$ from $\overline{q}_0$ to $\overline{q}_f$, there exists an interference-free path $T'(t) \in \mathcal{C}_{\text{approx}}$ from q_0 to q_f that is homotopic to it and whose path configurations are no farther than an ϵ distance from it:

$$\forall T(t) \in \mathcal{C}_{\text{free}} \ \exists T'(t) \in \mathcal{C}_{\text{approx}} \quad \max_{t \in [0,1]} \|T(t) - T'(t)\| \leq \epsilon$$

When the object shape approximations are conservative (i.e., the exact shape is a subset of the approximated shape), the configuration space approximation is also conservative ($\mathcal{C}_{\text{free}} \subset \mathcal{C}_{\text{approx}}$). Thus, the approximated path is guaranteed to be interference free for the exact shapes.

The complexity of the configuration space approximation (measured as the number of hyperplanes defining it) is related to the fit measure. A configuration space region with a fit measure of ϵ requires an approximation with resolution less than or equal to ϵ. The complexity of the approximated configuration space grows as ϵ decreases, since the original configuration space surface boundaries must be approximated by hyperplanes that are no farther than ϵ distance from them. Tight fits, which have small fit measures, require many hyperplanes. For example, the complexity of each configuration space cell in our algorithm is proportional to the number of body points and cavity facets, which ranges from 1,000 to 10,000 for clearances of 0.01 inch.

References

Barraquand, J., and Latombe, J. C. 1991. Robot motion planning: A distributed representation approach. *Int. J. Robot. Res.* 10(6):628–649.

Barraquand, J., Langlois, B., and Latombe, J. C. 1992. Numerical potential fields techniques for robot path planning. *IEEE Trans. Sys. Man and Cybernet.* 22.

Brooks, R. A., and Lozano-Pérez, T. 1983. (Karlsruhe, Germany). A subdivision algorithm in configuration space for find-path with rotation. *Proc. 8th Int. Joint Conf. on Artificial Intelligence.* Los Altos, NM: Morgan Kaufmann.

Buckley, S. J. 1990 (Scottsdale, AZ). Fast motion planning for multiple moving robots. *Proc. IEEE Int. Conf. on Robotics and Automation.* New York: IEEE.

Donald, B. 1987. A search algorithm for motion planning with six degrees of freedom. *Art. Intell.* 31(3):295–353.

Erdmann, M. 1994. On a representation of friction in configuration space. *Int. J. of Robot. Res.* 13(3):240–271.

Faverjon, B. 1984 (Atlanta). Obstacle avoidance using an octree in the configuration space of the manipulator. *Proc. IEEE Int. Conf. on Robotics and Automation.* New York: IEEE.

[2] We can define the clearance with metrics on Cartesian space as well.

Faverjon, B., and Tournassoud, P. 1987 (Raleigh, NC). A local-based approach for path planning of manipulators with a high number of degrees of freedom. *Proc. IEEE Int. Conf. on Robotics and Automation.* New York: IEEE.

Giraud, A., and Sidobre, D. 1987 (Raleigh, NC). A heuristic motion planner using contact for assembly. *Proc. IEEE Int. Conf. on Robotics and Automation.* New York: IEEE.

Gupta, K. K., and Zhu, X. 1994 (San Diego). Practical global motion planning for many degrees of freedom: A novel approach within a sequential framework. *Proc. IEEE Int. Conf. on Robotics and Automation.* New York: IEEE.

Kavraki, L., and Latombe, J. C. 1994 (San Diego). Randomized preprocessing of configuration space for fast path planning. *Proc. IEEE Int. Conf. on Robotics and Automation.* New York: IEEE.

Latombe J. C. 1991. *Robot Motion Planning.* Dordrecht; The Netherlands: Kluwer Academic Publishers.

Paul, H. A., Mittelstadt, B., Barger, W., Taylor, R., Kazanzides, P., Zohars, J., Williamson, B., and Hanson, W. 1992 (Nice, France). A surgical robot for total hip replacement surgery. *Proc. IEEE Int. Conf. on Robotics and Automation.* New York: IEEE.

Stulberg, S. D. 1989. Custom-made primary total hip replacements. *Orthopedics* 15(5):1245–1252.

Taylor, R. H., Paul, H. A., Kazanzides, P., Mittelstadt, B., Hanson, W., Zohars, H., Williamson, B., Musits, B., Glassman, E., and Bargar, W. 1994. An image-directed robotic system for precise orthopedic surgery. *IEEE Trans. Robot. Automation* 10(3):260–275 .