Hip Implant Insertability Analysis: A Medical Instance of the Peg-In-Hole Problem

Leo Joskowicz and **Russell H. Taylor**

IBM T.J. Watson Research Center
P.O. Box 704, Yorktown Heights, NY 10598, USA

Abstract

Recent advances in cementless hip replacement surgery have significantly improved the accuracy of bone cavity preparation and custom implant shape design. With the increased accuracy and tighter fit, determining whether the implant can actually be inserted to its final working position inside the cavity during surgery is essential to guarantee full implant functionality. This paper presents an implemented algorithm for determining if an implant can be successfully inserted into a cavity. The program computes an insertion trajectory consisting of small implant motion steps. Motion steps are computed by solving a series of linear optimization problems whose solution corresponds to the maximum allowed displacement in a preferred direction satisfying localized motion constraints. The program implements a greedy, search-free algorithm to find a monotone trajectory in a preferred direction to any desired resolution. The program has been successfully tested on real cases.

Introduction

Computer-based medical imaging and modeling systems, coupled with computer-aided design and manufacturing technology, have already begun to have a major impact on the clinical practice of medicine [12]. The design and fabrication of custom orthopedic implants from CT data is one growing application of such systems [10].

In the case of cementless implants, which rely on a press fit, or tissue ingrowth and significant surface-to-surface contact between the implant and the bone for fixation, the accuracy of bone preparation can have a significant effect on implant efficacy. Recent advances in robot bone machining have demonstrated an order-of-magnitude improvement in the accuracy of femoral canal preparation for hip replacement surgery [9, 11]. The resulting improved accuracy and consistency of bone preparation will greatly facilitate controlled studies of implant design efficacy. The added flexibility of robot machining, which can create shapes that are impractical to make using conventional hand-held instrumentation, presents both new possibilities and new problems for implant designers.

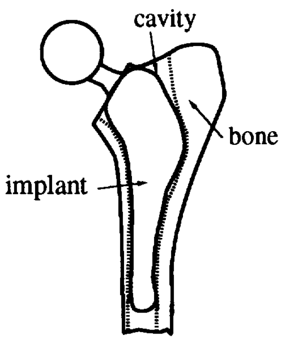

Figure 1: Sketch of a cementless hip implant tightly fit into a femur bone cavity

In cementless hip replacement surgery, the damaged joint connecting the femur and the hip is replaced by an implant tightly fit into the femur (Fig. 1). To install such implants, the surgeon typically starts by sawing off the femoral head and drilling a guide hole down the femur using a flexible reamer. The surgeon then drives a broach into the guide hole to make a cavity with the same shape as the implant. Since the broach shape matches the implant shape and the broach has been inserted into the cavity, there is some assurance that the implant will fit into the cavity carved by the broach. Unfortunately, the broach design may require the removal of bone which, for implant efficiency, would perhaps be better left intact. For example, the implant designer may desire the implant shape follow a (slightly smoothed out) iso-density contour, as determined from CT data. However, even if the robot can machine a cavity with the desired shape, there is no guarantee that the implant can actually be inserted into the cavity. It is highly undesirable discover this situation in the operating room. It would be much better to discover an insertability problem after the

1050-4729/93 $3.00 © 1993 IEEE

implant is designed and before it is fabricated. It is even more desirable to identify a feasible insertion trajectory automatically, if one exists, and to identify possible places for design modification if such trajectory cannot be found.

This paper presents an implemented algorithm for determining if an implant can be successfully inserted into a cavity. Given the shapes of the implant and the femoral cavity, the program computes a insertion trajectory for the implant which ensures, to any desired resolution, that the implant does not penetrate the cavity walls during the insertion. If the implant cannot be extracted, the trajectory terminates at a wedged implant configuration.

Related work

Testing for implant insertability is an instance of the classical path planning problem, where the goal is to find a collision-free trajectory of a moving object amidst fixed obstacles [8]. Finding a collision-free trajectory requires searching the space of non-overlapping object configurations (the configuration space) for a continuous path from the initial to the final configuration. *Global* strategies first construct a configuration space connectivity graph and then search it for the desired path. *Local* strategies directly search for the path, performing geometric computations as the search progresses.

Global methods require computing the configuration space, whose complexity is polynomial in the geometric size of the objects and exponential in their total number of degrees of freedom. The characteristics of the implant insertability problem – very complex 3D shapes, six degrees of freedom, tight fit, fine motion planning – rule out global strategies and their variations, such as hierarchical configuration space decomposition [3, 5], planning in low-dimensional configuration space projections [2], and exploiting the objects' geometrical regularities [7]. In addition, any strategy that repeatedly uses object overlap detection is impractical due to the objects' complexity.

Local strategies depend on the efficiency of the geometric computations and the effectiveness of the search strategy. Existing local strategies emphasize search effectiveness. Donald's [4] algorithm for a moving six degree of freedom polyhedron places a fine resolution grid on the configuration space and uses a set of heuristics based on the local configuration space geometry to search for the path. For a tight fit, this method requires a very fine grid resolution and precise geometric computations which significantly affect the overall efficiency. Other potential field methods [6, 1] make similar assumptions.

This paper presents a local path planning technique for the implant insertion problem that emphasizes efficient local geometry and motion constraints computation over search. Due to the tight fit and obvious general insertion direction, little or no search is necessary. However, the objects' complex shapes requires efficient local geometric computations. The method contributes to research in path planning by providing:

- a practical solution to a real problem, including a working program and tests on real data
- a formulation of approximated localized linear configuration space constraints for small six degree-of-freedom motions
- a formulation of linear programming problems to find the maximum allowed displacement in a preferred direction satisfying the motion constraints
- a greedy, search-free algorithm to find a monotone trajectory in a preferred direction to any desired resolution

Problem formulation

We formulate the implant insertion problem as a motion planning problem, where the implant is a three-dimensional moving object with six degrees of freedom, and the cavity is a fixed three-dimensional obstacle. A moving coordinate frame is attached to the origin of the implant. A fixed coordinate frame is attached to the origin of the cavity. The implant and cavity shapes are described with respect to their coordinate frames. The position and orientation of the implant in space, also called the *implant configuration*, is defined with respect to the cavity's fixed coordinate frame.

General formulation

Let $(\overline{p}, \overline{\theta})$ be the six configuration variables (three translations and three rotations) describing the configuration (position and orientation) of the implant with respect to the cavity's fixed coordinate frame[1]. Let $\overline{b}$ be the location of a point in the implant surface expressed in implant coordinates. Let $F(\overline{p}, \overline{\theta})$ be the transformation mapping points in implant coordinates to points in fixed coordinates in position $\overline{p}$ and orientation $\overline{\theta}$. The position $\overline{v}$ of an implant point $\overline{b}$ in configuration $(\overline{p}, \overline{\theta})$ with respect to the cavity's fixed frame is expressed as:

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

where $Rot(\overline{\theta})$ is the 3x3 rotation matrix specifying the orientation of the implant with respect to the cavity's fixed coordinate frame.

Let H be the function describing the shape of cavity. A point $\overline{x}$ lies on or inside the cavity when:

$$H(\overline{x}) \leq 0$$

[1]Notation conventions: lower case letters with an overbar, $\overline{x}$, denote three-dimensional vectors. Upper case letters with an overbar, $\overline{X}$, denote n-dimensional vectors. Bold capital letters, $\mathbf{A}$, denote matrices.

An implant point $\overline{b}$ in configuration $(\overline{p}, \overline{\theta})$ lies on or inside the cavity when:

$$H(F(\overline{p}, \overline{\theta}) \cdot \overline{b}) \leq 0 \qquad (1)$$

This inequality, formulated over the set of all implant points $\overline{b}$, defines the implant *configuration constraints* which must hold for the implant not to penetrate the cavity.

Let B be a set of points in the implant surface. The implant *configuration space* $\mathcal{C}$ is the set of implant configurations for which the implant and the cavity surfaces do not interpenetrate:

$$\mathcal{C} = \{(\overline{p}, \overline{\theta}) \mid H(F(\overline{p}, \overline{\theta}) \cdot \overline{b}) \leq 0, \ \forall \overline{b} \in B\}$$

We define an implant motion as a continuous function $T(t)$ specifying the position and orientation of the implant at time t. An implant trajectory $\boldsymbol{traj}(T(t))$ is the set of implant configurations reached during an implant motion. It is the image of the implant motion function.

An implant insertion motion is a feasible implant motion that takes the implant from an initial configuration outside the cavity to a final configuration inside the cavity. A motion is physically feasible iff the implant does not penetrate the cavity at any time during the motion. Thus, all implant configurations in a feasible implant trajectory must satisfy the implant configuration constraints in Eq. (1):

$$\boldsymbol{traj}(T(t)) = \{T(t) = (\overline{p}(t), \overline{\theta}(t)) \mid H(F(\overline{p}(t), \overline{\theta}(t)) \cdot \overline{b}) \leq 0, \ \forall \overline{b} \in B, \ t \in [t_0, t_f]\}$$

for $T(t_0) = (\overline{p}_0, \overline{\theta}_0)$ the initial configuration at starting time t_0 and $T(t_f) = (\overline{p}_f, \overline{\theta}_f)$ the final implant configuration at ending time t_f.

Note that the implant extraction trajectory – the implant trajectory that takes the implant from inside to outside the cavity – is identical to the implant insertion trajectory. The implant extraction motion is simply the implant insertion motion backwards in time. In the following, we refer to the implant insertion and extraction trajectories and motions interchangeably.

Problem specialization

Finding an exact solution to the implant insertion problem is computationally intractable and practically unnecessary. Instead, we solve an approximate version of the problem to a desired resolution by exploiting the properties of the problem and by making key approximations.

We compute the implant extraction trajectory because the implant's final configuration inside the cavity is known precisely, whereas the implant's initial configuration can be anywhere outside the cavity, above a prespecified height.

We decompose the problem into a series of simpler optimization problems, where the solution to each problem yields a fragment of the implant extraction trajectory. To decompose the extraction problem into simpler problems we:

- discretize and approximate the implant and cavity shapes
- linearize and simplify the configuration constraints
- discretize the extraction trajectory
- linearize small implant motions along the extraction trajectory

The resulting extraction trajectory configurations specify small implant motion steps that do not exceed a prespecified displacement and do not violate the configuration constraints.

The remaining of this section describes and motivates each assumption and approximation in detail. The next section describes the formulation of the optimization problems.

Because the implant and the cavity have relatively smooth shapes, we can describe their contours by a finite set of surface elements to any desired resolution. We discretize the implant shape by sampling its surface with control points $b_j \in B$. We discretize the cavity shape by decomposing its surface into small patches. Patches are defined around a set of points sampled on the surface of the cavity.

Let $\overline{x}_i$ be a sample point in the surface of the cavity and let $h_i(\overline{x})$ be the function describing the cavity patch in the neighborhood $\Delta(\overline{x}_i)$. The cavity shape $H(\overline{x})$ can be described by a family of simpler functions $h_i(\overline{x})$ such that:

$$\forall \overline{x} \ \exists h_i(\overline{x}) \text{ s.t. } (\overline{x} \in \Delta(\overline{x}_i) \ \wedge \ H(\overline{x}) = h_i(\overline{x}))$$

The neighborhood $\Delta(\overline{x}_i)$ can be chosen, for example, to be the set of points contained in the sphere centered at $\overline{x}_i$ of radius r_i.

Because of the cavity's smooth shape, we can approximate the cavity patches with polynomial functions to any desired resolution. In particular, we can approximate the cavity patches with planar facets $\overline{a}_i.\overline{x} - \overline{c}_i$ such that:

$$\mid h_i(\overline{x}) - (\overline{a}_i.\overline{x} - \overline{c}_i) \mid \leq \ \boldsymbol{res}, \ \forall \overline{x} \in \Delta(\overline{x}_i)$$

for a given resolution $\boldsymbol{res}$. We jointly describe the cavity facet and its neighborhood by a planar convex cell $\mathbf{A}_i.\overline{x} - \overline{C}_i$. A point $\overline{x}$ is inside the cell when:

$$\mathbf{A}_i.\overline{x} - \overline{C}_i \leq \overline{0} \qquad (2)$$

Having discretized the implant shape with a set of control points $\overline{b}_j \in B$, we can rewrite the implant configuration constraint in Eq. (1) as a collection of configuration constraints on each implant control point:

$$H(F(\overline{p}, \overline{\theta}) \cdot \overline{b}_j) \leq 0$$

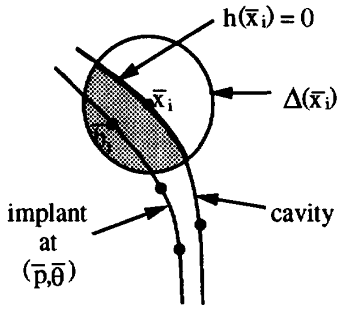

Figure 2: Detail of an implant control point $\overline{b}_j$ in configuration $(\overline{p},\overline{\theta})$ in the neighborhood of cavity point $\overline{x}_i$. The shaded area corresponds to $h(\overline{x}_i) \leq 0$.

Because of the tight fit between the implant and the cavity, a motion of each implant control point is constrained by a small piece of the cavity's surface in its immediate neighborhood (see Fig. 2). That cavity patch imposes configuration constraints that subsume the configuration constraints imposed by all other patches, provided the control point remains in the neighborhood of the patch. For any implant configuration in the neighborhood of $(\overline{p},\overline{\theta})$, there exists, for every implant control point $\overline{b}_j$, a patch $h_i(\overline{x})$ such that:

$$h_i(F(\overline{p},\overline{\theta})\cdot\overline{b}_j) \leq 0 \iff H(F(\overline{p},\overline{\theta})\cdot\overline{b}_j) \leq 0 \quad \text{for } F(\overline{p},\overline{\theta})\cdot\overline{b}_j) \in \Delta(\overline{x}_i)$$

Thus, for each implant configuration in the extraction trajectory, it is sufficient only to consider the configuration constraints imposed by the patches that are nearest to each implant control point. The inequality

$$h_i(F(\overline{p},\overline{\theta})\cdot\overline{b}_j) \leq 0 \qquad (3)$$

defines the *local* configuration constraint of implant control point $\overline{b}_j$.

Since the implant and cavity shapes are relatively smooth, the extraction trajectory will also be relatively smooth. We discretize the implant extraction trajectory $traj(T(t))$ with a sequence of configurations $(\overline{p}_k,\overline{\theta}_k)$ such that $(\overline{p}_0,\overline{\theta}_0)$ and $(\overline{p}_m,\overline{\theta}_m)$ are the initial and final implant configurations and:

$$(\overline{p}_{k+1},\overline{\theta}_{k+1}) = (\overline{\epsilon}_k,\overline{\alpha}_k) \oplus (\overline{p}_k,\overline{\theta}_k)$$

where $(\overline{\epsilon}_k,\overline{\alpha}_k)\oplus(\overline{p}_k,\overline{\theta}_k)$ is defined by $F(\overline{\epsilon}_k,\overline{\alpha}_k)\cdot(\overline{p}_k,\overline{\theta}_k)$ and $(\overline{\epsilon}_k,\overline{\alpha}_k)$ are small configuration displacements that do not exceed a prespecified maximum displacement: $(\overline{\epsilon}_{max},\overline{\alpha}_{max})$:

$$\begin{aligned} |\,\overline{\epsilon}_k\,| &\leq \overline{\epsilon}_{max} \\ |\,\overline{\alpha}_k\,| &\leq \overline{\alpha}_{max} \end{aligned}$$

We can now derive the discretized approximate trajectory $traj(T_D(k))$ by substituting the configuration constraints of Eq. (3) into the trajectory definition of Eq. (2), obtaining:

$$\begin{aligned} traj(T_D(k)) = \{&T_D(k) = (\overline{p}_k,\overline{\theta}_k) \mid \\ &h_i(F(\overline{p}_k,\overline{\theta}_k)\cdot\overline{b}_j) \leq 0,\ \ \forall \overline{b_j} \in B,\ 0 \leq k \leq m\} \end{aligned}$$

$T_D(0)$ and $T_D(m)$ are the initial and final configurations of the implant.

We have thus reduced the problem of finding an extraction trajectory to finding a sequence of implant configuration displacements $(\overline{\epsilon}_k,\overline{\alpha}_k)$ which satisfy the configuration constraints and do not exceed a predefined maximum displacement.

Extraction steps

We compute each $(\overline{\epsilon}_k,\overline{\alpha}_k)$ iteratively by formulating a series of linear optimization problems. Initially, we set $\overline{p}_0$ and $\overline{\theta}_0$ to the initial position and orientation of the implant. We compute the maximum displacement $(\overline{\epsilon}_k,\overline{\alpha}_k)$ in a preferred direction that satisfies all the configuration constraints, maintains all the implant control points in their neighborhood, and is less than the maximum allowed displacement. We then move all the implant control points to the new configuration $(\overline{p}_{k+1},\overline{\theta}_{k+1}) = (\overline{\epsilon}_k,\overline{\alpha}_k) \oplus (\overline{p}_k,\overline{\theta}_k)$ and repeat the process until the final configuration is reached.

We now describe the formulation of the linear optimization problem for step $k+1$ using the result of step k. Let $\overline{v}_{jk}$ be the position of point b_j in configuration $(\overline{p}_k,\overline{\theta}_k)$:

$$\overline{v}_{jk} = F(\overline{p}_k,\overline{\theta}_k)\cdot\overline{b}_j \qquad (4)$$

The new position $\overline{v}_{jk+1}$ of an implant control point b_j in configuration $(\overline{p}_{k+1},\overline{\theta}_{k+1})$ is obtained by displacing b_j in configuration $(\overline{p}_k,\overline{\theta}_k)$ by $(\overline{\epsilon}_k,\overline{\alpha}_k)$:

$$\begin{aligned} \overline{v}_{jk+1} &= F(\overline{\epsilon}_k,\overline{\alpha}_k)\cdot\overline{v}_{jk} \\ &= Rot(\overline{\alpha}_k)\cdot\overline{v}_{jk} + \overline{\epsilon}_k \end{aligned}$$

Since $(\overline{\epsilon}_k,\overline{\alpha}_k)$ is a small configuration displacement, we can approximate it by:

$$\overline{v}_{jk+1} \simeq (\overline{\alpha}_k \times \overline{v}_{jk}) + \overline{v}_{jk} + \overline{\epsilon}_k$$

for $|\,\overline{\epsilon}_k\,| \leq \overline{\epsilon}_{max}$ and $|\,\overline{\alpha}_k\,| \leq \overline{\alpha}_{max}$.

We can rewrite the local configuration constraints in Eq. (3) as:

$$\begin{aligned} h_i(\overline{v}_{jk+1}) &\leq 0 \\ h_i((\overline{\alpha}_k \times \overline{v}_{jk}) + \overline{v}_{jk} + \overline{\epsilon}_k) &\leq 0 \end{aligned}$$

Substituting the planar facet approximation of $h_i(\overline{x})$ from Eq. (2) we obtain:

$$\mathbf{A}_i\cdot((\overline{\alpha}_k \times \overline{v}_{jk}) + \overline{v}_{jk} + \overline{\epsilon}_k) - \overline{C}_i \leq 0$$

$$(\overline{v}_{jk} \times \mathbf{A}_i)\cdot\overline{\alpha}_k + (\mathbf{A}_i\cdot\overline{\epsilon}_k) \leq \overline{C}_i - \mathbf{A}_i\cdot\overline{v}_{jk} \qquad (5)$$

for every $b_j \in B$, $|\bar{\epsilon}_k| \leq \bar{\epsilon}_{max}$, and $|\bar{\alpha}_k| \leq \bar{\alpha}_{max}$. We have thus obtained a family of linear constraints in $(\bar{\epsilon}_k, \bar{\alpha}_k)$ that approximate the local configuration constraints.

We find the largest displacement $(\bar{\epsilon}_k, \bar{\alpha}_k)$ in a preferred direction by setting and solving the linear optimization problem LP_k:

$$\begin{array}{rcl} & \textit{maximize } \tau(\bar{\epsilon}_k, \bar{\alpha}_k) & \\ & \text{subject to} & \\ (\bar{v}_{jk} \times \mathbf{A}_i) \cdot \bar{\alpha}_k + (\mathbf{A}_i \cdot \bar{\epsilon}_k) & \leq & \overline{C}_i - \mathbf{A}_i \cdot \bar{v}_{jk} \\ |\bar{\epsilon}_k| & \leq & \bar{\epsilon}_{max} \\ |\bar{\alpha}_k| & \leq & \bar{\alpha}_{max} \end{array} \qquad (6)$$

where the objective function $\tau(\bar{\epsilon}_k, \bar{\alpha}_k)$ is a linear function that locally maximizes the extraction displacement in the preferred direction. Since the primary preferred direction is the cavity's vertical axis z, we chose, for example, $\tau(\bar{\epsilon}_k, \bar{\alpha}_k) = \epsilon_z$.

Implant insertion/extraction algorithm

We now present the implant extraction algorithm. The extraction trajectory is a sequence of small implant configuration displacements in a preferred direction satisfying the local configuration constraints in Eq. (3). The configuration displacements are iteratively computed by solving a series of linear programming problems LP_k (Eq. (6)). Solving the linear programming problem LP_k yields the largest small configuration displacement consistent with the configuration constraints formulated in the neighborhood of the implant control points.

Moving the implant by the maximum allowable displacement at each step in a preferred direction is a greedy strategy to reach the final configuration without searching for alternative trajectories. It is successful when monotone progress towards the final implant configuration along the preferred direction is possible. This is usually the case for typically smooth implant and cavity shapes. The implant's preferred extraction direction is parallel to the cavity's vertical axis. We expect the extraction trajectory to be mainly vertical displacements compensated by small variations in the other motion parameters. Note that monotone progress in a preferred direction is a desirable property because it facilitates the manual insertion of the implant into the cavity. Alternatively, we can temporarily select a different preferred direction to "unwedge" the implant. We have not found this to be necessary in the cases we have studied.

Solving the linear programming problem LP_k yields a small configuration displacement which defines the new non-overlapping implant configuration in the extraction trajectory. When moving the implant to its new configuration in preparation for the next step, one of the following scenarios occurs:

1. the implant has moved by the largest small displacement step
2. some implant control points have reached their neighborhood boundaries
3. the implant did not move

If the implant has moved by the largest small displacement step allowable, the extraction can proceed by finding a new displacement from the new configuration. The linear programming problem LP_{k+1} is formulated by recomputing the same local configuration constraints of LP_k in the new configuration. However, if some implant control points have reached their neighborhood boundaries, any further displacement in the preferred direction will take these points out of their neighborhoods, thereby invalidating their local configuration constraints. A transition to adjacent neighborhoods is necessary to continue the extraction. The new linear programming problem LP_{k+1} is formulated by finding the new implant control points neighborhoods, replacing their configuration constraints in LP_k, and computing the resulting configuration constraints in the new configuration. Finally, if the implant did not move, it is stuck in a wedged position and no further motion in the preferred direction is possible. We can either decide that the extraction has failed, or pick a different preferred direction by changing the objective function $\tau(\bar{\epsilon}_k, \bar{\alpha}_k)$ and resolving LP_k.

The solution of LP_k provides the information necessary to determine which scenario occurs and to formulate LP_{k+1}. The basis of linear programming problem LP_k indicates which inequality constraints are active (an inequality constraint is said to be active when 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 displacement values indicate that no displacement is possible in the preferred direction.

To formulate the local configuration constraints for LP_k (Eq. (5)), we keep track of the cavity neighborhoods $\Delta(\bar{x}_i)$ each implant control point b_j is in configuration $(\bar{p}_k, \bar{\theta}_k)$. The local configuration constraints are then formulated by computing the values of $\bar{v}_j k$ (Eq. (4)) for all the implant control points $\bar{b}_j$ in configuration $(\bar{p}_k, \bar{\theta}_k)$ and substituting the appropriate cavity patch description.

The extraction algorithm is summarized in Table 1. First, we sample the implant surface with a set of control points and partition the cavity into a set of small planar facets. Next, we establish the initial correspondence between implant control points and cavity facets in the initial configuration and formulate the initial linear programming problem LP_0. We then solve LP_0

```
1. select implant control points
2. approximate cavity with planar facets
3. match implant control points and cavity facets in
   initial configuration
4. set $(\overline{\epsilon}_0, \overline{\alpha}_0)$ to initial configuration, and $k$ to 0
5. formulate the initial problem $LP_0$
6. while implant is below desired height do
   (a) solve $LP_k$ to obtain displacement $(\overline{\epsilon}_k, \overline{\alpha}_k)$
   (b) if the displacement is null,
       - declare the implant stuck and fail, or
       - replace the objective function and
         re-solve $LP_k$.
   (c) set next implant configuration $(\overline{p}_{k+1}, \overline{\theta}_{k+1})$ to
       $(\overline{\epsilon}_k, \overline{\alpha}_k) \oplus (\overline{p}_k, \overline{\theta}_k)$
   (d) if one or more small displacement constraint is
       active,
       formulate $LP_{k+1}$ by computing the configuration
       constraints in new implant configuration
   (e) if one or more configuration constraints are
       active,
       formulate $LP_{k+1}$ by replacing the new configuration
       constraints and computing them in the
       new implant configuration
   (f) increment $k$ by 1
7. return the extraction trajectory $(\overline{p}_k, \overline{\theta}_k)$
```

Table 1: Implant extraction algorithm

and move the implant to its new configuration by the specified displacement. If the implant did not move, it cannot be extracted in the preferred direction. Depending on the amount of search and backtracking desired, we can try again by choosing a new preferred direction. If the the implant has moved by the largest small displacement step, we formulate the next step by recomputing the local configuration constraints at the new implant configuration. If one or more implant control points have reached their neighborhood boundary, we find their adjacent cavity neighborhoods and formulate the next step by replacing their configuration constraints and computing the new set of local configuration constraints at the new implant configuration. We repeat this process until the implant reaches a desired height outside the cavity (the z position of the implant is greater than a prespecified height). The resulting displacement configuration sequence defines the motion steps that extract the implant from the cavity.

Implementation

We have implemented the implant extraction algorithm. The inputs to the program are the implant and cavity shapes described as data extracted from catscan (CT) images, the initial position of the implant inside the hole, and the desired implant and cavity resolutions. The output of the program is an extraction trajectory and a graphical animation of the extraction. If no extraction trajectory is found, the program stops at the wedged implant configuration.

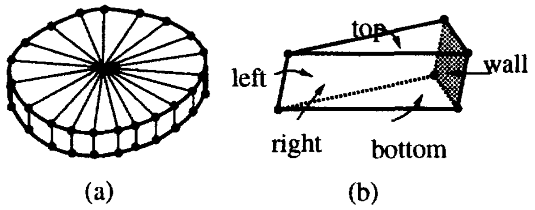

Figure 3: Cavity discretization: (a) partition of two consecutive contours into cells; (b) a single cell.

The program is written in C; it uses IBM's optimization subroutine library (OSL) to solve each linear optimization problem LP_k and the graphics library GL to display three-dimensional views of the implant extraction sequence. This section provides some implementation details.

The implant and cavity shapes are defined by a stack of parallel, two dimensional contour slices described by spline segments. The implant control points $\overline{b}_j$ are selected by sampling the contour splines of each implant slice at the desired resolution.

The cavity is discretized to a desired resolution by partitioning it into disjoint cells. Each cell describes a cavity facet and its neighborhood. Cells are created by dividing contour slices with vertical planes centered around the cavity's spine (Fig. 3(a)). Cells are pie-slice shaped volumes consisting of five planar faces: top, bottom, right, left, and cavity wall (Fig. 3(b)). The wall face, corresponding to a portion of the cavity's inner wall is defined by fitting an approximating plane through four points, two on the upper contour slice, two on the lower contour slice. Each cell $cell_i$ is defined by a set of five planes, $\mathbf{A}_i.\overline{x} - \overline{C}_i \leq 0$. A cell mesh is produced by recording cell adjacency relationships (left, right, top, bottom) between neighboring cells.

We add a very wide and high "lip" slice to the top of the cavity, so that implant control points always lie inside a cavity cell. The lip dimensions are such that the implant is guaranteed never to touch lip walls.

We maintain a data structure that records the cavity cell that each implant control point is in. We establish the initial correspondence between implant control points $\overline{b}_j$ and cavity cells $cell_i$ in the initial configuration $(\overline{p_0}, \overline{\theta_0})$ by testing which cell inequalities hold:

$$\mathbf{A_i} \cdot (F(\overline{p_0}, \overline{\theta_0}).\overline{b}_j) - \overline{C}_i \leq 0$$

Batch	*Geometry*			*Trajectory*		*CPU*
	res.	growth	size	out?	steps	*time*
test1	0.1	0.05	4,638	yes	1,830	863
test1	0.1	0.02	4,638	no	535	321
test1	0.05	0.04	9,405	yes	5,323	4,560
test2	0.1	0.05	19,345	yes	2,340	6,328
test2	0.1	0.02	19,345	no	657	2,345
screw	0.1	0.01	1,630	yes	483	435

Table 2: Experiment results.

We update the correspondence as the implant moves by using the cell adjacency relationships between cells.

We formulate a linear programming problem LP_k by computing the local configuration constraints as described in the previous section and adding the small displacement constraints. For n implant control points and m cavity cells, this yields $5n.m + 12$ inequalities in 6 variables. For efficiency reasons, we solve the dual problem, not the problem itself.

(In an earlier version of the program, we found the cell migration process too slow. To compensate, we create an overlapping cavity mesh, so simultaneously migration places implant point well inside cavity cells.)

Experiments

We have tested our program on a number of real and synthetic examples. Fig. 4 shows a typical example of a real implant extraction sequence.

Typical implants are 4" high with cross section diameters between 0.5" and 2". The spacing between CT cross-sections is either 0.1" (test1) or 0.05" (test2). Each cross section requires between 25 and 80 splines to describe it. To describe the implant shape at a resolution of 0.05" (maximum distance between consecutive control points in a slice) takes about 20,000 control points. The cavity shape matches closely the implant shape and is often its exact complement grown by about 0.05". In most cases, the critical implant configurations in the extraction trajectory are near the initial configuration. At about 3" height, the implant can be directly extracted from the cavity by a straight vertical motion. We chose a corkscrew with elliptical cross-section as a synthetic example to demonstrate the performance of the program when no straight vertical translation is possible. In all cases, the largest small translational displacement limit is 0.1" and the largest small rotational displacement limit is one degree.

Table 2 summarizes runs on several examples. We tested three batches with different dimensions and at different resolutions (all dimensions are in inches). The first three columns describe the implant and cavity geometry: the resolution at which the implant and body shapes were sampled, the amount by which the implant was grown to obtain the cavity, and the total number of points required to describe the shapes. The following two columns indicate if the implant can be extracted successfully and the number of motion steps required. The last column indicates the required CPU time in seconds on an IBM RS/6000 model 530.

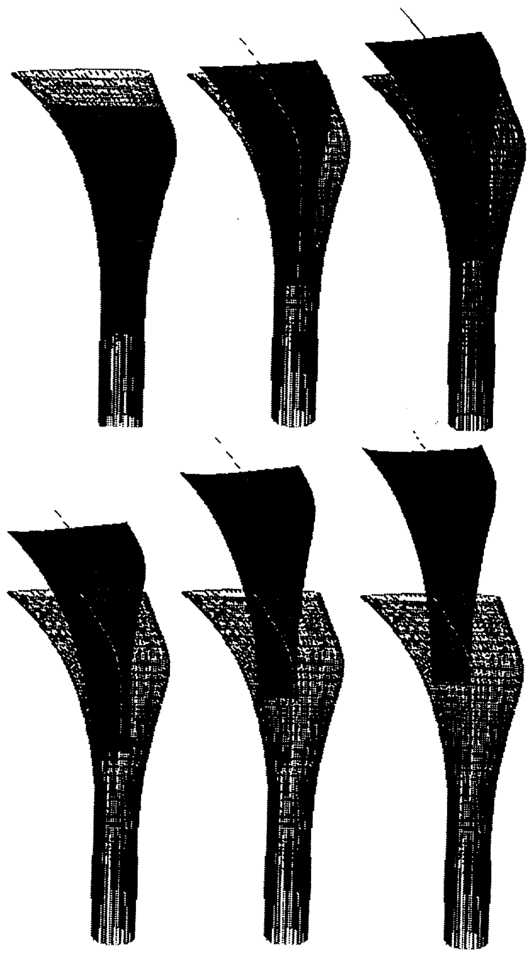
Figure 4: An extraction sequence of a real implant

Conclusions and future work

We have presented an implemented algorithm for determining if an implant can be successfully inserted into a cavity. The program computes an insertion trajectory consisting of small implant motion steps. Motion steps are computed by solving a series of linear optimization problems whose solution corresponds to the maximum allowed displacement in a preferred direction satisfying localized motion constraints. The program implements a greedy, search-free algorithm to find a monotone trajectory in a preferred direction to any desired resolution.

We are considering a number of alternative methods that will speed up the implant extraction by increasing the size of the steps in the extraction trajectory, thus reducing the total number of steps required to extract the implant. For example, we can force an (adaptively) small implant motion step in a preferred direction and use the implant configuration constraints to check that the implant is still inside the cavity, or otherwise reposition it to satisfy those constraints. We are also exploring localized search strategies for wedged situations, which will *guarantee* finding an extraction trajectory in any direction when such a trajectory exists. An important but difficult extension is to take into account friction during the extraction.

Automatic implant insertability analysis opens up a host of possibilities for supporting and automating implant design from CT data. To begin with, the implant designer can explore more design modifications and alternative solutions knowing that implant insertability can be tested quickly and efficiently. In addition, the information produced by the insertability tests can be used to modify designs. When a particular implant design cannot be extracted, the test identifies the implant's furthest wedged position. Displaying the implant in that position allows the designer to identify the parts of the implant design should be modified to allow the insertion.

The insertability test can also serve as the basis for automatic implant shape modification and redesign. Starting with an initial implant and cavity shape, the goal is to achieve the tightest insertable fit by modifying the implant's dimensions and shape. A simple (but not very good) method is to shrink or grow the implant uniformly by a prespecified amount until the tightest insertable fit is achieved (this is the method we used in our experiments). A better method is to only shrink the implant surface in contact with the cavity walls when the implant is wedged, or slightly shorten the implant's stem, or else to make similar modifications to the cavity shape. In the long term, the goal is to couple shape constraints with computable criteria expressing the underlying goals of the implant design process. Many of these criteria are derived from physical properties (density, etc.) obtainable from CT data. One can imagine a design process in which processing on the CT data set produces an "ideal" implant shape (with respect to the bone's physical properties), which is then modified by considering constraints generated by insertability and manufacturability analysis, by engineering analysis of forces and function, and by suggestions from the human designer.

Finally, we note that our insertion/extraction method can be applied to problems with similar characteristics in other domains, where a complex-shaped object moves in a highly constrained and complex environment. Two examples of such domains are mechanical component assembly/disassembly and casting mold design and removal.

Acknowledgments

We thank John Amber and Craig Blaschke from Biomet, Inc. for introducing us to the problem and providing us with CT implant data. We thank David Jensen and the other members of OSL at IBM Research, Alan Kalvin, and Gabriel Taubin for advice on the implementation.

References

[1] J. Barraquand and J.C. Latombe, "Robot Motion Planning: A Distributed Representation Approach", Stanford Tech. Report STAN-CS-89-1257, May 1989.

[2] S.J. Buckley. "Fast Motion Planning for Multiple Moving Robots", *Proc. of IEEE Int. Conf. on Robotics and Automation*, Scottsdale 1990.

[3] R.A. Brooks and T. Lozano-Perez, "A Subdivision Algorithm in Configuration Space for Find-Path with Rotation", *Proc. of the 8th Int. Joint Conference on Artificial Intelligence*, Karlsruhe, FRG, 1983.

[4] B. Donald, "A Search Algorithm for Motion Planning with Six Degrees of Freedom", *Artificial Intelligence*, **31**-3, 1987.

[5] B. Faverjon, "Obstacle Avoidance Using an Octree in the Configuration Space of the Manipulator", *Proc. of IEEE Int. Conf. on Robotics and Automation*, Atlanta 1984.

[6] B. Faverjon, P. Tournassoud, "A Local-Based Approach for Path Planning of Manipulators with a High Number of Degrees of Freedom", *Proc. of IEEE Int. Conf. on Robotics and Automation*, Raleigh, 1987.

[7] A. Giraud, D. Sidobre, "A Heuristic Motion Planner Using Contact for Assembly", *Proc. of IEEE Int. Conf. on Robotics and Automation*, Raleigh, 1987.

[8] Jean-Claude Latombe, *Robot Motion Planning*, Kluwer Academic Publisher, 1991.

[9] H. Paul *et al.*, "A Surgical Robot for Total Hip Replacement Surgery", *Proc. of IEEE Int. Conf. on Robotics and Automation*, Nice, France 1992.

[10] S.D. Stulberg, "Custom-made Primary Total Hip Replacements", *Orthopedics*, **15**-5, 1989.

[11] R.H. Taylor *et al.*, "An Image-based Robotic System for Hip Replacement Surgery", *Journal of the Robotics Society of Japan*, pp 111-116, 1990.

[12] J. Udupa and G. Herman, *3D Imaging in Medicine*, CRC Press, Boca Raton, 1991.