A Complete Algorithm for Designing Planar Fixtures Using Modular Components

Randy C. Brost, *Member, IEEE,* and Kenneth Y. Goldberg, *Member, IEEE*

Abstract— Commercially available modular fixturing systems typically include a lattice of holes with precise spacing and an assortment of precision locating and clamping modules that can be rigidly attached to the lattice. Currently, machinists manually design a suitable arrangement of these modules to hold a given part. This requires expertise and can delay production. Futhermore, a machinist may in many cases settle upon an arrangement that is not optimal for a given machining operation.

In this paper we present an implemented algorithm that accepts a polygonal description of the part silhouette, and efficiently constructs the set of all feasible fixture designs that kinematically constrain the part in the plane. Each fixture is comprised of three locators rigidly attached to the lattice and one sliding clamp, and constrains the part without relying on friction.

The algorithm is based on an efficient enumeration of admissible designs that exploits part geometry and a graphical force analysis. The algorithm run time is linear in the number of designs found, which is bounded by a polynomial in the number of part edges and the part's maximal diameter in lattice units. Our review of the literature suggests that this is the first fixturing algorithm that is *complete* in the sense that it is guaranteed to find all admissible fixture designs for an arbitrary polygonal part silhouette and to identify the optimal fixture relative to an arbitrary quality metric. The algorithm does not consider out-of-plane forces or motions; however, we view this planar result as an essential component of a larger algorithm that solves the 3-D fixture design problem by treating the planar and out-of-plane constraint problems separately. This approach is analogous to the widely used 3-2-1 fixture design heuristic, and appears to be applicable to a broad class of man-made parts.

I. INTRODUCTION

MOST automated manufacturing, assembly, and inspection operations require fixtures to locate and hold parts. Given part shape and desired position and orientation, fixtures are usually custom designed by manufacturing engineers and machinists. Although there are a few general guidelines and a number of studies, systematic algorithms for automatically designing fixtures based on CAD part models are still lacking [1], [2].

This is partly due to the uncountable set of alternative fixture designs that must be considered in the general case. One way

to reduce the number of alternatives is to limit consideration to a small set of components that must be located on a regular lattice structure. Such *modular* fixturing systems also have the advantage of allowing rapid set-up and changeover for new parts, precision locating on a tightly toleranced lattice, and a reduced fixture inventory comprised of re-usable components [3].

The concept of modular fixturing using a family of interchangeable components was originally developed in England during World War II, and has resulted in a variety of commercially available modular fixturing systems [4]. These systems typically include a square lattice of tapped and doweled holes with spacing toleranced to ±0.0002 in and an assortment of precision locating and clamping elements that can be rigidly attached to the lattice using dowel pins or expanding mandrels. Although the lattice and set of modules greatly reduce the number of alternatives, designing a suitable fixture currently requires human intuition and trial-and-error. Designing a new fixture can be time consuming. Furthermore, if the set of alternatives is not systematically explored, the designer may settle upon a suboptimal design or fail to find any acceptable design.

In this paper we present an algorithm for automatically designing a class of modular fixtures. These fixtures constrain all motion of a part in the support plane. Constraint is provided by four point contacts and does not rely on friction. Each fixture in this class uses three round *locators*, each centered on a lattice point, and one translating *clamp* that must be attached to the lattice via a pair of unit-spaced holes, thus allowing contact at a variable distance along the principle axes of the lattice. We use the term *fixel* (fixture element) to refer to either a locator or a clamp and the term *fixture* to refer to a geometric arrangement of three locators and one clamp on the lattice.

An acceptable fixture design must satisfy several requirements. First, it must fully constrain the part to prevent its motion. We require fixtures to provide *form closure,* which is a kinematic constraint condition that prevents all motion [5]. In addition to constraining the part, the fixture must not interfere with certain geometric regions of the part, perhaps due to cosmetic surfaces or the need to retain clearance for grasping, machining, assembly, or other operations. Thus we define *geometric access constraints,* which define regions of points that must remain free of fixture components. With these requirements in mind, we say that a fixture is *admissible* if it provides form closure and obeys the geometric access constraints. In this paper, we further restrict our attention to fixtures where each fixel makes point contact with only one

linear edge of the part. Given a part as input, the algorithm efficiently enumerates all admissible fixtures and ranks them according to a user-definable scalar quality metric.

The algorithm begins with a geometric transformation that expands the part edges by the radius of the locators; this allows us to treat the locators as points on the transformed edges. These edges are then trimmed to respect geometric access constraints. We define a *locator setup* as the combination of three locator positions and a part configuration such that the part contacts all three locators. The algorithm enumerates all locator setups; for each, it identifies the set of all points along the perimeter of the object where an additional contact would provide form closure constraint. This in turn allows us to identify all of the possible form-closure clamp locations for each locator setup; each clamp location defines a unique fixture design. After pruning this set of fixtures by checking for geometric violations such as the clamp body intersecting the part or other fixels, we rank the surviving admissible fixtures based on some scalar metric such as ability to resist an applied force without exerting large contact reaction forces. The resulting fixtures are then returned to the user.

We believe this is the first modular fixture design algorithm that is *complete* in the sense that it is guaranteed to find an admissible fixture if one exists. It is essential to acknowledge that such a fixture does not always exist; for example, parts that are much smaller than the lattice spacing will have no available fixture design. Other parts for which no fixture exists are considered in [6].

This algorithm is guaranteed to find the *optimal* fixture. Since the algorithm constructs the set of all fixture designs possible for a given modular fixturing kit, the algorithm can score the constructed designs according to a user-supplied quality metric, sort the results, and return the fixture with the highest score. The contact-force analysis included in our implementation is one example of such a metric.

There are a number of issues that are not considered by the algorithm. For example, out-of-plane motions and part deformations are both important considerations in some fixturing problems, and these are not addressed by our planar fixture design algorithm. However, the strong analysis and enumerative aspects of our algorithm make it well-suited for use as part of a larger procedure that synthesizes 3-D fixture designs for prismatic parts, which occur in a variety of man-made products. Limitations and possible extensions of this algorithm are discussed further in Section V.
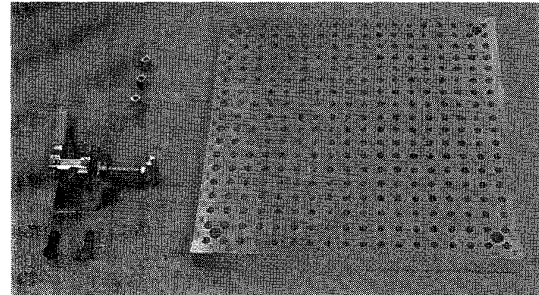
A. Example

Fig. 1 shows an example. The part is shown on the left of Fig. 1(a). This plastic housing is one-half of the case of a commercially available hot glue gun. We would like to design a fixture to hold this housing (the part) while assembling the gun. In Fig. 1(b) the part boundary is represented as polygon, as are geometric access constraints that delineate regions that must remain clear of fixture components—in this case to allow the gun tip, trigger, and cord to be assembled with the part. The part has 28 edges, and the access constraints have a total of 20 edges.
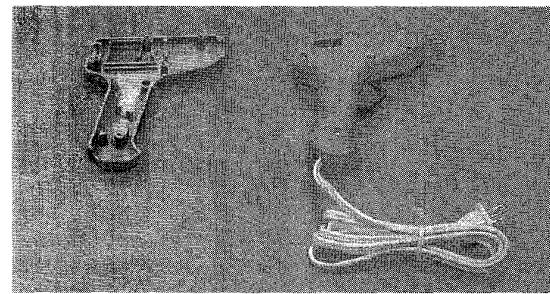
(a)
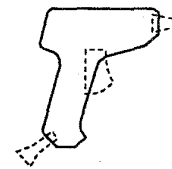
(b)

(c)
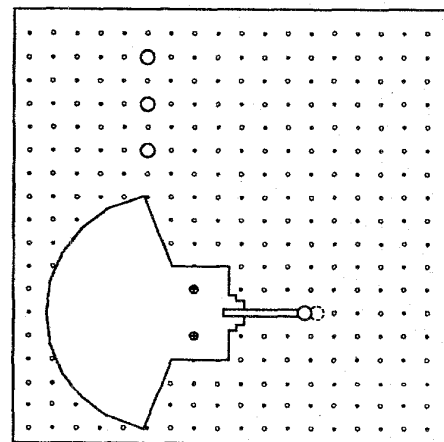
(d)

Fig. 1. An example. (a) The part to fixture is shown at the left. (b) Polygons describing the part and geometric access constraints (shown dashed) which must remain free of fixture components. (c) A modular fixturing kit from Qu-Co, Inc. (d) Our model of the fixture kit. Filled dots represent tapped holes, and open dots represent dowel holes. The dashed outline shows the clamp extension limit.

A commercial modular fixturing kit is illustrated in Fig. 1(c), comprised of a precisely machined plate with alternating dowel/threaded holes, a set of three round locators,

and a manually actuated clamp. In Fig. 1(d) this kit is represented by a model of the plate, the locators, and the clamp. The clamp is modeled by a polygon delineating the space occupied by the clamp during normal operation; this includes the region swept as the handle is moved from the open to closed position. The clamp model also includes a polygon describing the shape of the clamp plunger, and the plunger travel limits.

For this example, the algorithm returned 97 fixture designs in just over two minutes on a desktop workstation, sorting them according to a quality metric which examines the maximum contact force required to resist an arbitrary unit applied force. Fig. 2 shows three of the returned fixture designs. Note that all three fixture designs provide form closure and obey the geometric access constraints.

If we consider a clockwise unit torque applied to the part, we see that the fixture in Fig. 2(a) is superior to the one in Fig. 2(c), where contacts A and B must exert very large contact reaction forces to resist the torque. Our implementation includes these considerations in the metric it uses to rank fixtures; see Section III-F. Fig. 3 shows the design of Fig. 2(a) assembled using the fixture kit.

II. PREVIOUS WORK

There is a substantial literature on fixturing and the related topic of grasping. These results may be roughly grouped into three categories: fundamental analysis of the existence of fixtures or grasps, analysis of a given fixture or grasp, and automatic synthesis of fixtures or grasp configurations.

A. The Existence of Fixtures and Grasps

The century-old definition of *form closure* captures the intuitive function of a fixture [5]. A set of contacts provides form closure if infinitesimal part motion is completely constrained; equivalently, the set of frictionless contacts is able to resist arbitrary forces and torques on the part. This condition may be analyzed using the concept of a *wrench*, which is a generalized force that includes moment contributions [7]. In the plane, a set of wrenches provides form closure if they positively span $\Re^3$.

Reuleaux showed that at least four wrenches are necessary for form closure in the plane [5]. Recently, Markenscoff *et al.* showed that four wrenches are sufficient for any piecewise-smooth compact connected planar body, excluding surfaces of revolution [8]. For objects in 3-D space, it is known that seven wrenches are necessary for form closure [9], [10]; Markenscoff *et al.* showed that seven wrenches are sufficient for polyhedra [8].

The above proofs demonstrate the existence of form-closure fixtures where contacts may occur at arbitrary positions in space. In the case of modular fixtures, the locations of contacts are constrained by the modular fixturing kit. Mishra studied the problem under these constraints, and showed that a fixture can always be found for a rectilinear part as long as all edges have length of four or more lattice units [11]. Zhuang *et al.* showed that for the modular fixture kit employed in this paper, there exist polygons of arbitrary size for which no fixture design exists [6].
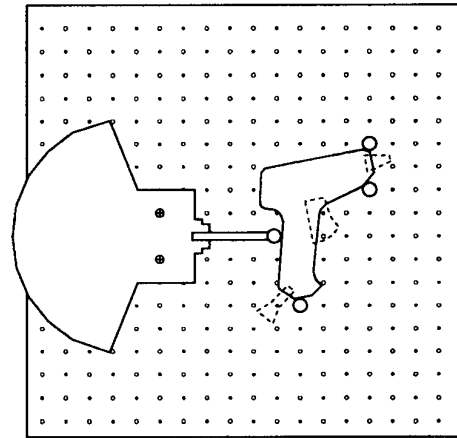
(a)
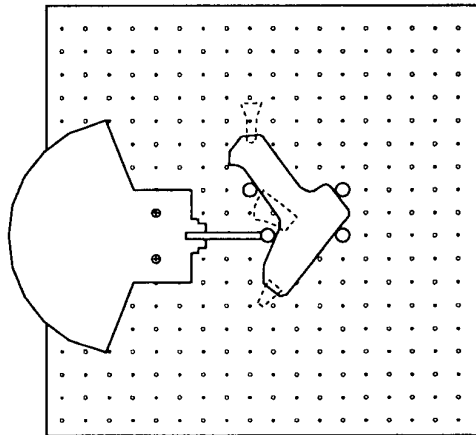
(b)
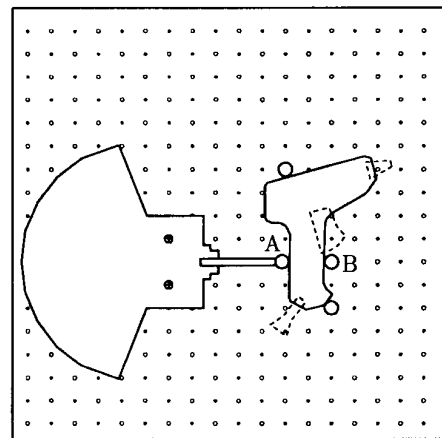
(c)

Fig. 2. Three admissible fixtures designed by the algorithm. In each, the part is fixtured by three round locators and one translating clamp aligned with the lattice. For this example, our implementation designed 97 fixtures in 129.4 s.

B. Fixture/Grasp Analysis

Given a fixture or grasp configuration, a variety of considerations may be applied to evaluate the suitability of the fixture or grasp for a given task. In this section, we review
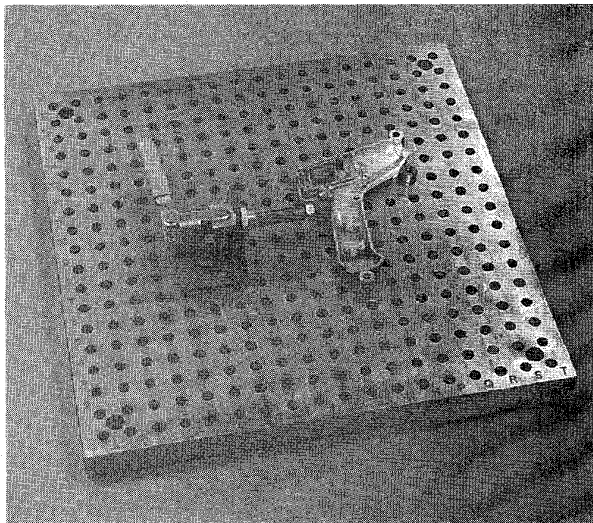
Fig. 3.   The assembled fixture of Fig. 2(a).

past work on evaluating fixtures and grasps; some of these evaluation criteria are already embodied in our algorithm (such as the determination of whether or not the fixture provides form closure), while others would be appropriate to include in future extensions to the algorithm's quality metric.

Asada and By showed how to determine whether a given fixture design provides total constraint of a rigid body, as well as loading accessibility before clamping [12]. Our glue gun example is inspired by their example of a power-drill housing, and highlights the similarities and differences between the results. Our paper extends Asada and By's analysis methods by providing an automatic design procedure that considers geometric access constraints and modular assembly constraints in addition to kinematic closure. However, our algorithm considers only three degrees of freedom.

Several grasp quality measures have been proposed based on the smallest contact force necessary to resist applied forces [13], [14]. Such a metric can be defined as the solution to an optimization problem [15], [16] or geometrically as the radius of the largest sphere that can be embedded inside the wrench convex [17], [18]. One subtlety is that the wrench space is not homogeneous, and one must take care when comparing forces with torques. In a similar vein, Bausch and Youcef-Toumi developed a method of evaluating the degree of motion constraint imposed by seven fixels contacting a three-dimensional rigid body [19]. The quality metric we describe in Section III-F is similar to Bausch and Youcef-Toumi's metric, except that our metric analyzes planar problems and explicitly considers expected task forces. Unlike some previous geometric quality metrics, our quality metric is not sensitive to the selection of a force/torque scaling factor, which can be a somewhat arbitrary parameter.

Other authors addressed task-specific requirements that must be satisfied by a fixture. Englert reported methods for assessing a fixture design's susceptibility to part deformation, locator wear, and dynamic chatter during machining operations [20]. Sakurai later explored the relationship between fixture design, part tolerances, and part deformation in greater detail [21].

Sakurai also studied the relationship between cutting forces and clamping forces in fixtures that rely on friction for part restraint—particularly when top-clamps are employed. Lee and Cutkosky extended Sakurai's results by clarifying the relationship between a fixture with top clamps and the friction limit surfaces that were previously developed to study the motion of sliding planar bodies [22]. Kim further extended these results by considering whether or not expected force magnitudes would exceed clamp stress limits [23]. Other authors reviewed additional practical requirements of fixture designs [24], [25], [1]. The algorithm we report in this paper is complementary to these results; while we do not present an analysis of part deformation, tolerances, or maximum allowable clamp forces, it is reasonable to expect that these and other considerations could be folded into the quality metric that rates fixture designs. Ultimately, we envision that the synthesis methods of this paper could be combined with enhanced quality metrics to produce a larger system that will select the fixture that best satisfies this myriad of considerations.

Schimmels and Peshkin examined the problem of loading a given fixture using generalized damper compliant motions, and showed that in the absence of friction, a robust loading strategy exists for all deterministic fixture designs [26]. (The fixture designs returned by our algorithm are always deterministic in their sense.) Later work characterized the conditions where a fixture may be robustly loaded if friction is present [27]. Schimmels and Peshkin considered only infinitesimal motions, and did not analyze the effect of finite motions such as large rotations. Whether or not a fixture may be reliably loaded is an important aspect of fixture design that we do not treat in this paper; this consideration may be used either to discard fixtures that cannot be loaded robustly, or as a metric to compare fixtures based on their ease of loading.

C. Fixture/Grasp Synthesis

For a review of current practice in manual fixture design, see [28]. In addition to these methods, a number of techniques have been developed for automatically synthesizing grasp and fixture configurations.

Mishra, Schwartz, and Sharir described an algorithm for synthesizing form-closure grasps on an arbitrary 2-D or 3-D object; the grasps returned by the algorithm may require up to six contacts for 2-D objects [29]. Nguyen gave an algorithm for finding a set of four (seven) independent regions on the boundary of a polygon (polyhedron) such that a frictionless contact applied to each region is guaranteed to provide form closure [30]. Such regions are useful because they allow for uncertainty in the part's pose. For three frictional contacts in the plane, Ponce and Faverjon showed that comparable regions on a polygon could be found using linear optimization [31]. These results both allow contacts at arbitrary points in space, and do not consider the constraints imposed by a modular fixture kit.

In the context of planning grasping strategies for lifting an object off a supporting surface, Trinkle and Paul identified jamming regions on an object's boundary, given three point contacts [32]. These jamming regions are analogous to the form-closure clamping regions produced by our force-sphere

analysis, in that they identify portions of the object boundary where an additional contact would lead to a force-closure condition. The form-closure analysis presented in this paper differs from Trinkle and Paul's construction in that it applies to all possible arrangements of contact normals, while Trinkle and Paul's construction addresses the special case where two of the contact normals are parallel.

In the specific context of fixturing, Mani developed a method for designing planar fixtures based on Reuleaux's rotation center analysis techniques [33]. Given a polygonal part shape, Mani's procedure identifies all topologically equivalent fixture designs. However, his procedure does not accurately consider the fixel shape or the mapping of the fixture design onto a modular fixture plate. Chou et al. developed a procedure that designs fixtures for prismatic parts using screw algebra, geometric heuristics to place locators at positions that allow easy loading of the part, and a rotation center analysis to place clamps [34]. As with Mani, they did not consider the constraints imposed by a modular fixture kit. Kim developed a procedure that designs fixtures using top clamps [23]. The procedure focuses primarily on placing the top clamps and estimating the required clamping force; lateral locators are only allowed on user-specified orthogonal datum surfaces, which are assumed to be aligned with the hole grid. Our work complements Kim's result, since we generate all possible locator placements for an arbitrary part shape but do not address top clamping. Kim also developed a procedure for designing fixture setups using a vise. In related work, Hayes and Wright developed an expert system for planning machining operations [35], [36]. This system analyzed the interaction between machined features and constructed a sequence of setup plans that would allow a part to be machined from raw stock while avoiding feature interaction problems (such as drilling into a slanted surface). This system employed a simplified feature-based geometric analysis to design fixtures; it may be possible to extend the scope of this high-level planning system by including the more detailed geometric analysis presented in this paper. Finally, Englert and Sakurai also reported fixture design methods based on geometric heuristics [20], [21]; these methods do not have the generality of the algorithm presented here.

Recently, Wallack and Canny reported an algorithm for designing a class of modular fixtures with four round locators on a split lattice that can be closed like a vise [37]. Their algorithm, like ours, takes the part shape as input and enumerates all combinations of fixture elements that achieve form closure. Also, like ours, their algorithm sweeps edges to compute contact conditions and runs in polynomial time. However, the algorithms differ in the construction of the fourth fixel location. In the case of Wallack and Canny's split vise, the third fixel's (x, y) position may not be known until the location of the fourth fixel is chosen, at which time the part pose may be determined. Consequently their algorithm includes another nested loop instead of the direct force-sphere construction we employ. Further, their algorithm does not require a check for interference between the part and the clamp body. The net result of these differences is that their algorithm entails one less factor of n and one more factor of d in its asymptotic

complexity (see Section III-G), while providing the additional capability of designing fixtures with two translating fixels instead of just one.

III. THE ALGORITHM

A. Problem Statement

Assumptions.
- Parts and locators are rigid solids. A part can be represented with a simple polygon, and locators can be represented as circles with identical radii less than half the grid spacing l ($\frac{\sqrt{2}}{2}l$ on an alternating grid). Thus we do not have to check collisions between locators.
- All contacts are ideal unilateral point constraints. Our analysis treats these contacts as frictionless: the fixtures do not depend on any minimum level of friction.

The algorithm only generates fixtures where each fixel contacts the interior of a single part edge. Thus we neglect fixtures where a fixel contacts a part vertex or multiple part edges. Further, we treat all fixtures that can be mapped onto each other through translation and/or rotation as equivalent, and only generate one fixture from each equivalence class.

Input.
- Polygonal part boundary, provided as a list of vertices.
- A set of geometric access constraints, provided as a list of polygons defined in the part coordinate frame.
- Height and width of the fixture plate lattice.
- Locator radius.
- Description of the clamp. This includes a polygon describing the shape of the clamp body, locations of the clamp mounting holes, a polygon describing the shape of the clamp plunger, and its min/max travel limits. The tip of the plunger is assumed to be a circle of the same radius as the locators.
- A quality metric. This is a function that accepts a fixture design and returns a scalar quality measure.

Output.
A list of all admissible fixtures for the part, sorted in order of quality.

B. Overview of the Algorithm

Given the input described above, the algorithm produces its output by performing the following steps.
1) The input is transformed by growing the part such that the fixels can be treated as ideal points, and the fixture plate lattice is assumed to be infinite.
2) All possible candidate fixture designs are constructed. This is accomplished by enumerating the set of possible locator setups, and then passing the result to a form-closure analysis routine that constructs all of the possible abstract clamp locations for each setup. Each locator setup and clamp location specifies a unique fixture.
3) The set of candidate fixtures are then filtered to remove those that do not obey clamp travel limits, cause collisions with the clamp body or plunger, or do not fit on the finite fixture plate.
4) The resulting fixtures are scored according to the user-specified quality metric, and then sorted in order of

decreasing score. The algorithm returns the sorted list of fixtures.

The following sections will explain each of these steps in detail.

C. Transforming the Input

The first step of the algorithm is a transformation that allows us to treat round locators as ideal points. This is accomplished by forming the Minkowski sum of the polygonal part boundary and the circular fixel shape; fixturing the resulting expanded boundary with ideal points is then equivalent to fixturing the original part boundary with finite-radius locators. Thus it is sufficient to consider points on the edges of this expanded boundary as candidate positions for locators.

Although the expanded boundary has rounded edges corresponding to contacts between a locator and an object vertex, we consider only the linear components of the expanded boundary. We similarly grow the constraint regions by the fixel radius, and then restrict our attention to the subset of the expanded part edges which do not intersect the grown constraints. This will assure that the fixels of all generated fixtures will avoid the access constraint regions. This results in a list of rigidly attached but possibly unconnected linear edges. See Fig. 4. We are now free to translate and rotate this group of edges to bring edges into contact with lattice centers.

D. Generating Candidate Fixtures

We proceed to enumerate all possible fixtures. First, we enumerate triplets of locators, identifying the part configurations consistent with each. Each combination of a locator triplet and an (x, y, θ) configuration specifies a locator setup. After enumerating all possible locator setups, we identify the set of all clamp positions that provide form closure.

1) Enumerating Locator Triplets: To enumerate all locator triplets, the following steps are repeated for all combinations of three edges, where either all three edges differ, or two of the three edges are identical. For example, (e_1, e_5, e_2) and (e_4, e_7, e_4) are both valid edge combinations. Order is not significant, so there are $\binom{n}{3} + 2\binom{n}{2}$ such combinations for n edge segments. The second combinatorial is multiplied by two because there are two valid triples for every choice of two distinct edges. Note that we need not consider combinations with three identical edges, since a part with three locators on one edge cannot be held in form closure.

Given a combination of three edges, (e_a, e_b, e_c), we can assume without loss of generality that e_a makes contact with a locator at the origin of the lattice. By translating and rotating e_a about the origin, e_b sweeps out an annulus centered on the origin, with inner diameter equal to the minimum distance between e_a and e_b and outer diameter equal to the maximum distance between e_a and e_b. That is, for any orientation of e_a, as we translate along the extent of e_a, e_b sweeps out a parallelogram. The union of these parallelograms as we rotate e_a forms an annulus. To eliminate equivalent fixtures, we only need to consider the first quadrant of this annulus. See Fig. 5.

We now consider each of these second locator positions in turn, and identify all possible positions for the third locator.
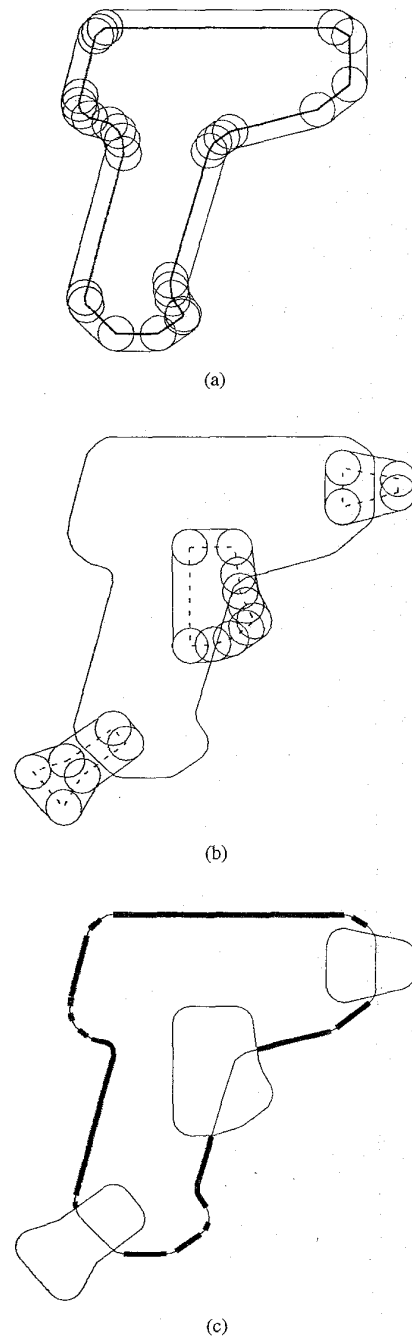

(a)

(b)

(c)

Fig. 4. Transforming the input. (a) Growing the part by the fixel radius. (b) After growing the constraints by the fixel radius. (c) The resulting edge segments.

If the first locator contacts e_a and the second locator contacts e_b, then a third locator in contact with e_c must be pairwise consistent with both e_a and e_b. The exact region swept out by e_c as we maintain contact with the first two locators is difficult to characterize. However, we can easily find an envelope that contains this region by independently considering each pair. That is, the possible locations for e_c with respect to e_a form an annulus around the origin, and the possible locations for e_c with respect to e_b form an annulus around the second locator.
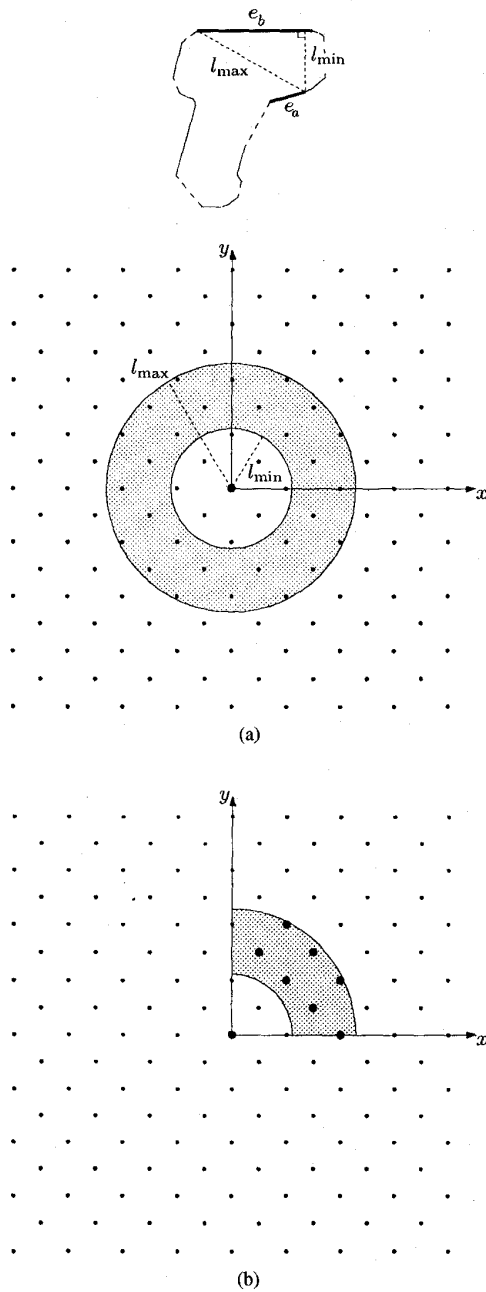
Fig. 5. Identifying candidate locations for locator #2. (a) The set of points swept by edge e_b, while e_a maintains contact with the origin. (b) The candidate locations for locator #2, after removing symmetries. Note that points on the y-axis must be excluded.

Intersecting these annuli provides a conservative bound on the set of grid locations that simultaneously satisfy both constraints.

We can further refine this bound by considering the angular limits for each annulus. This is accomplished by first identifying the angular limits of the part configurations that simultaneously contact the first and second locators, producing a $[\theta_{\min}, \theta_{\max}]$ interval of reachable part angles. Then we transform this interval by adding the $[\beta_{\min}, \beta_{\max}]$ interval that

delineates the minimum and maximum angle attainable by a ray connecting e_a and e_c. The resulting $[(\theta_{\min} + \beta_{\min}), (\theta_{\max} + \beta_{\max})]$ interval describes the set of all possible angles between points on edge e_a and e_c, while e_a and e_b maintain contact with locators 1 and 2. This interval defines a sector of the $e_a e_c$ annulus; points outside this sector are unreachable by e_c. A similar construction produces a sector of the $e_b e_c$ annulus based on the β-interval corresponding to edges e_b and e_c. Intersecting these annular sectors provides a set of candidate locations for the third locator. See Fig. 6.

2) Identifying Consistent Part Configurations: For each triplet of locators and associated contact edges, we must identify the set of consistent part configurations. This is accomplished by a configuration-space analysis that constructs the intersection points of edge/vertex–edge/vertex (ev–ev) contact equations. This calculation identifies intersection points between the ev–ev edges on the configuration-space obstacle corresponding to two-point contact situations. For example, if e_a, e_b, and e_c are the edges of the part in contact with fixels v_1, v_2, and v_3 respectively, then the combinations $e_a v_1 - e_b v_2$, $e_a v_1 - e_c v_3$, and $e_b v_2 - e_c v_3$ all correspond to two-contact situations that have an associated one-dimensional locus of points in the (x, y, θ) configuration space. Three-point contact is only possible at the intersections of these loci, so the set of part configurations where all three fixels are in contact may be found by solving for the roots of the parametric equations describing these intersections. This analysis is further discussed in Appendix A.

There may be up to two solutions to these equations, corresponding to different poses of the part that permit simultaneous contact with the three chosen locators (see Fig. 7). In these cases, we generate two candidate locator setups, one for each pose. In certain geometric situations there are an infinite number of solutions (such as when all three edges are parallel); these cases are discarded because they do not constrain the part to a unique location.

3) Enumerating Clamp Configurations: So far we have enumerated all possible three-edge combinations, all possible locator triplets for each edge combination, and all possible part configurations for each locator triplet. This has produced a list of all possible locator setups for the part. Next, we visit each setup and generate all of the possible clamp positions that provide form closure. Thus for each setup, we may generate several candidate fixtures, each with a different clamp position.

To generate the set of form-closure clamp positions for a locator setup, we perform a constraint analysis on the *force sphere*, a unit sphere centered at the origin of the $(F_x, F_y, \tau/\rho)$ space of planar forces.

This representation was previously described in [38]; see [39] for implementation details. This space represents both the direction and moment components of a line of force exerted in the plane. For example, Fig. 8 shows an example planar force and its corresponding point on the force sphere; the equations describing this mapping are presented in Appendix B. Note that if we were performing dynamic analysis, we would choose the part origin and ρ to correspond to the the part center of mass and radius of gyration; however, in our purely static analysis these may be chosen arbitrarily.
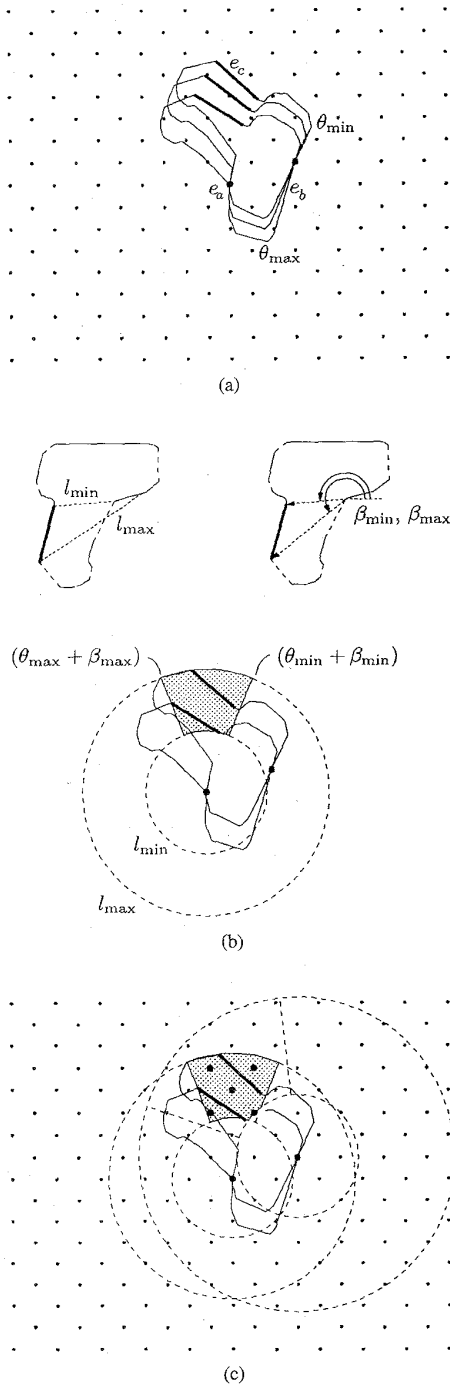
(a)

(b)

(c)

Fig. 6.   Identifying candidate locations for locator #3.

We treat each fixel/edge contact as an ideal unilateral point constraint. Thus each fixel may resist motion by exerting a reaction force in the direction of the inward-pointing contact normal. Fig. 9 shows the set of points on the force sphere corresponding to the three contact normals of a typical locator setup. The convex-combination of these points is also shown; this triangle on the force sphere delineates the set of all total contact reaction forces that may be produced by combining forces from all three contacts.
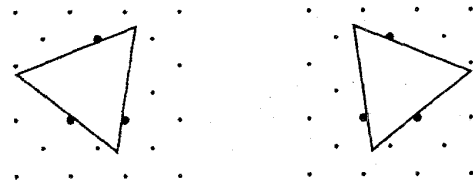

Fig. 7.   Two poses of a part that permit simultaneous contact with three locators.
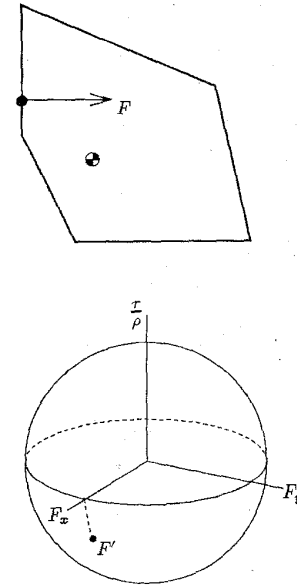
Fig. 8.   Mapping a line of force onto the force sphere.
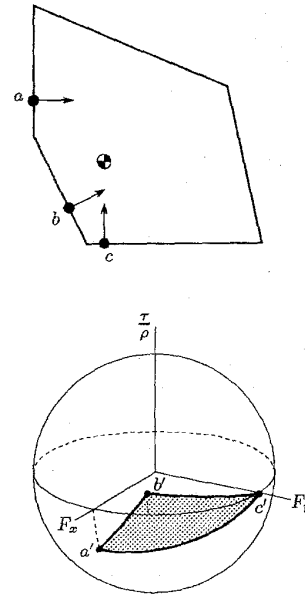

Fig. 9.   Constructing the set of reaction forces that may be generated by a locator setup. Each contact normal is mapped onto the sphere; the resulting points define a triangle of possible reaction forces.

A fixture design provides form closure exactly when the corresponding set of contact normals positively spans the entire force sphere. When this condition is satisfied, combinations of

contact reaction forces may produce an arbitrary total reaction force, thus opposing an arbitrary motion. Put another way, if the set of contact normals for a given fixture design span the force sphere, then all possible motions will violate at least one kinematic constraint.

Given a set of three contact normals corresponding to a locator setup, we can directly construct the set of forces that would produce form closure if provided as a fourth contact normal. This is accomplished by forming the convex-combination of the three contact normals on the force sphere, and then centrally projecting this triangle onto the opposite side of the sphere. The resulting negated triangle delineates the set of all forces that will produce form closure. If we can find a clamp position with a contact normal that corresponds to a point strictly in the negated triangle, then this clamp position and the three locators will define a form-closure fixture.

We can directly construct the set of clamp positions that satisfy this condition. We accomplish this by characterizing the set of all contact reaction forces that can be applied by a contact along the perimeter of the grown part. This set of forces is illustrated in Fig. 10. Note that the set of all possible contact forces corresponds to a "zig-zag" locus of points that encircle the force sphere. Fixel contacts along the edges of the polygon correspond to the vertical edges of the locus; note that as a force moves along an edge, only the torque component of its wrench will vary. Fixel contacts with the vertices of the polygon correspond to the diagonal locus edges. By intersecting the vertical locus edges with the set of possible form-closure forces constructed previously, we can identify the set of all available edge-contact normals that produce form closure for a given locator setup. We then map this set of contact normals back onto the grown part perimeter to identify the regions where a fourth contact point will produce form closure. Finally, we identify the set of possible clamp positions by intersecting the identified regions with the horizontal and vertical edges of the fixture lattice. This construction is illustrated in Fig. 11.

E. Filtering the Candidates

At this point the algorithm has enumerated all form-closure fixtures where the round fixels obey the geometric access constraints. The next step is to filter the candidates through several geometric tests. First, we determine the clamp location and check clamp travel limits. Next, we discard those fixtures where the clamp body or plunger intersects the part, the locators, or the access constraints. Finally, we attempt to fit the remaining fixtures on the finite fixture plate; fixtures that cannot be placed either horizontally or vertically are also discarded.

F. Ranking the Survivors

The final step of the algorithm is to rank the surviving fixtures according to the user-supplied quality metric. A user may then view the top fixtures and apply additional criteria to select a winner.

Our implementation includes a default quality metric that favors fixtures that can resist expected applied forces without generating excessive contact reaction forces. Large contact
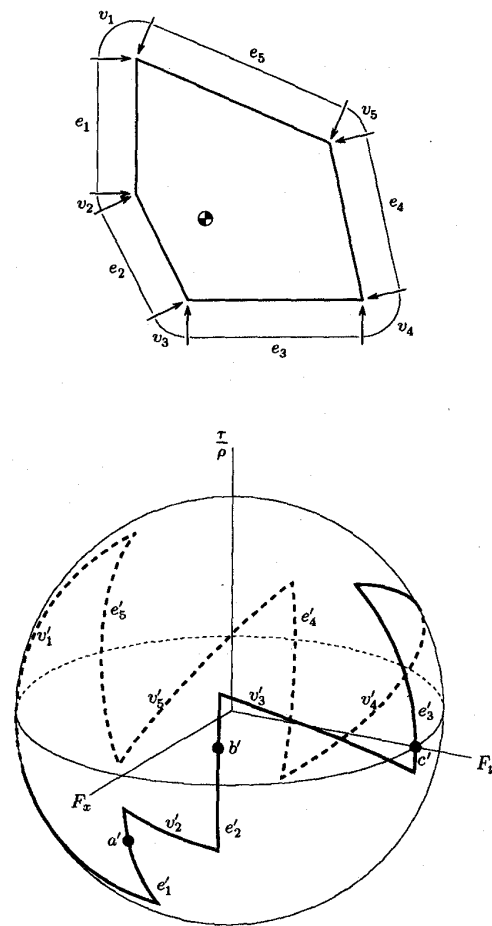

Fig. 10. The "zig-zag" locus of all forces that may be exerted by contacting the part at a single point. The contact normals of Fig. 9 are shown for reference. This construction previously appeared in [38], and an analogous mathematical definition appeared in [29].

forces are undesirable because they may deform the part. The effect of fixture geometry on contact reaction force is illustrated in Fig. 12. In this figure, a part is held in two different fixtures, both of which provide form closure. Which fixture is better? The answer depends on the forces that will be exerted on the part. For example, if downward forces will be applied to the part, then fixture A is better than fixture B, since fixture B will develop large "wedging" forces between the fixels. On the other hand, if clockwise torques will be applied, then fixture B is superior, since fixture A must develop large contact reaction forces to oppose rotation of the part.

As an example, we implemented a quality metric that allows the user to specify a list of expected forces on the part. These forces are represented by a list of force-sphere regions with associated magnitudes that could arise from operations such as machining, assembly, or pallet transfer operations. The quality metric scores each fixture by estimating the maximum contact reaction force required to resist the list of expected applied forces.

The estimated maximum contact reaction force for a given fixture is calculated by visiting each force-sphere region in the applied force list, generating a discrete sampling of points
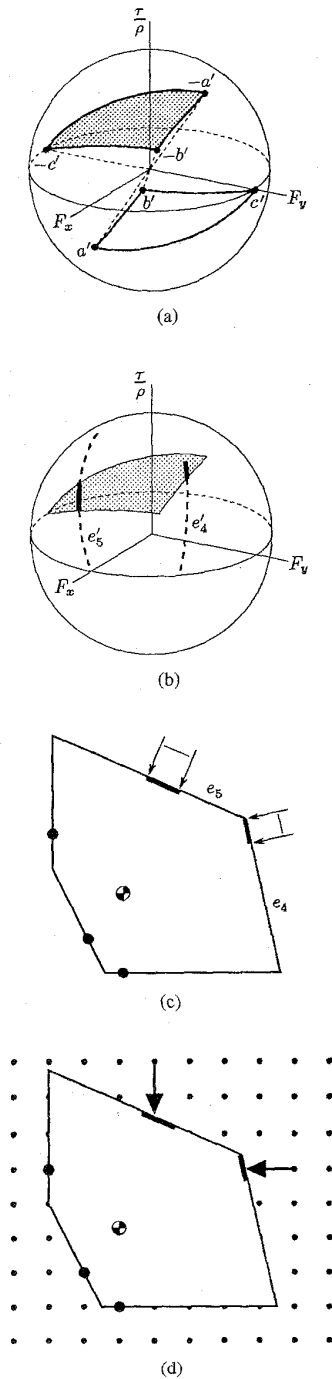
(a)

(b)

(c)

(d)

Fig. 11. Constructing the set of form-closure clamp placements. (a) Negating the set of possible contact reaction forces. (b) Intersecting the result with the vertical edges of the zig-zag locus. Contacts corresponding to the bold arcs will produce form closure. (c) The bold segments indicate the set of all edge-contact points that produce form closure. (d) Possible clamp placements.

in the region, computing the maximum contact reaction force required to resist each point, scaling the result by the associated magnitude, and taking the maximum of all the resulting contact reaction forces.

Given a particular point p' within a force-sphere region, the maximum contact reaction force may be constructed directly.
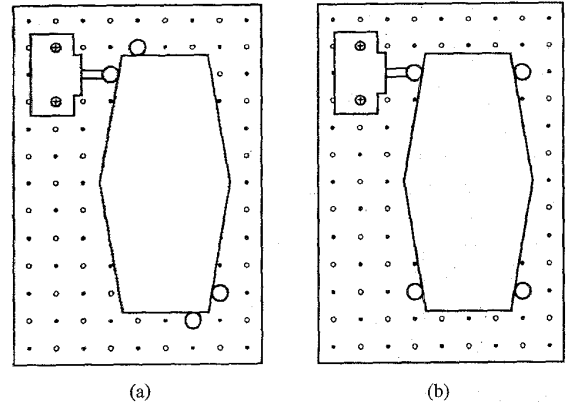

(a)   (b)

Fig. 12. Two fixture designs: (a) Fixture A and (b) Fixture B. Which one is better? The answer depends on the forces that are expected.

First the negation of the point $-p'$ is constructed. Since the four force-sphere points corresponding to fixel contact normals positively span the force sphere, the point $-p'$ must lie in a triangle formed by three of the normals, along an edge formed by two normals, or exactly coincide with one normal. In each of these cases, $-p'$ may be expressed as a positive linear combination of the corresponding normals, and the associated scaling factors may be computed directly. These scaling factors determine the magnitude of each contact reaction force in the force space; projecting the resulting scaled vector onto the $[F_x, F_y]$ plane produces the contact reaction force in the real space. The maximum contact reaction force then corresponds to the force with the largest magnitude $\sqrt{F_x^2 + F_y^2}$.

G. Algorithm Complexity

An asymptotic upper bound on the running time of the algorithm can be derived as follows. For the given polygonal part, let n be its number of edges and d the length of its maximum diameter (in units of lattice spacing). The enumeration considers $O(n^3)$ triplets of edges. For each triplet of edges, there are $O(d^2)$ locations for the second locator since we consider a sector of an annulus of diameter no greater than the part, and similarly for each pair of locators, there are $O(d^2)$ locations for the third locator. Once the part pose is determined by three locators, the number of possible clamp locations is bounded by its perimeter: $O(nd)$. Thus the maximum number of possible fixtures is $O(n^4 d^5)$. Checking for unwanted collisions can be accomplished in $O(n)$ time for each fixture, since the number of clamp edges is constant. If the quality metric can also be evaluated in $O(n)$ time or less for each fixture, then the total running time for the algorithm is $O(n^5 d^5)$.

IV. IMPLEMENTATION RESULTS

We have implemented the algorithm in Common Lisp on a Symbolics XL-1201 Lisp Machine, and run the program on several example problems. One example is shown in Fig. 1; in this problem the part has 28 edges, and the maximum diameter of the part is 5.3 in. The geometric access constraints are described by three polygons with a total of 20 edges. The fixture plate is an 18×18 array of alternating dowel/threaded

holes, with a spacing of 0.75 in. The fixel radius is 0.25 in. The clamp and plunger have 23 and 4 edges, respectively. For this problem, the algorithm produced 23 grown part edges, 416 candidate fixtures, and 223 admissible fixtures; this computation took 294.4 s.

Reviewing the output of the program suggests a number of simple heuristics for improving performance by eliminating poor fixtures early in the computation. We implemented some of these heuristics by providing three user-specified control parameters:

- *maximum clamp angle:* The maximum allowable angle between the clamp travel axis and the contact normal. Large clamp angles are undesirable because they can produce large contact forces, and may lead to binding of the clamp plunger.
- *edge trim distance:* After growing the part edges by the fixel radius and removing the segments that intersect the geometric access constraints, the resulting segments are further shrunk by this distance. This prevents fixel contacts very close to a part vertex; these contacts are undesirable because part vertices are weaker than edges, and may have inaccurate shape models due to chamfers or rounded corners.
- *minimum clamp clearance:* The minimum allowable distance between the clamp body or plunger and the part or geometric access constraints.

Each of these heuristics could be implemented through the quality metric by including penalties for large clamp angles, etc. Instead, our program includes these parameters in the construction procedure, thus avoiding the work required to build and analyze fixtures that would eventually be given poor quality scores. For our examples we chose 45° for the maximum clamp angle and 0.04 in (1 mm) for the edge trim distance and minimum clamp clearance. This produced a significant performance improvement, reducing the computation time for the example in Fig. 1 to 129.4 s. With these parameters, the program produced 17 grown part edges, 162 candidate fixtures, and 97 final fixtures.

Examining a more complex example reveals a number of interesting points. Fig. 13 shows a housing used to hold electronic components; in production this housing is fabricated by casting a near-net shape part, and then machining the casting to produce the fine details. We would like to design a fixture to hold the part during these machining operations. We modeled the part as a polygon with 57 edges, with a maximum diameter of 9.3 in. A small constraint region was included to account for a cut-out on one wall of the part. Given this input, our program designed 5770 fixtures in 3.3 h. Fig. 13(b) of the figure shows the design with the highest score for our default quality metric; if we had specified expected machining forces, a different design would have produced the highest score.

A number of issues spring to mind. First, do we really need to consider 5770 designs for this problem? Examining the list of designs reveals that many fixtures have very similar quality scores. This suggests that producing a subset of these designs may still lead to a very good fixture. From the above complexity analysis, we can observe that the large number of
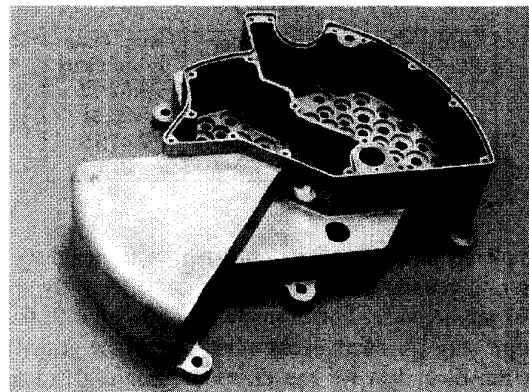
(a)
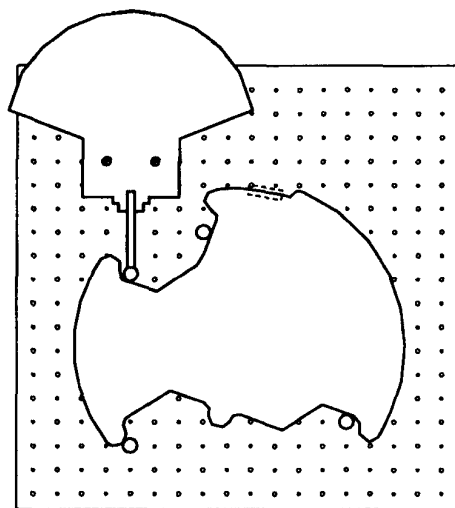
(b)
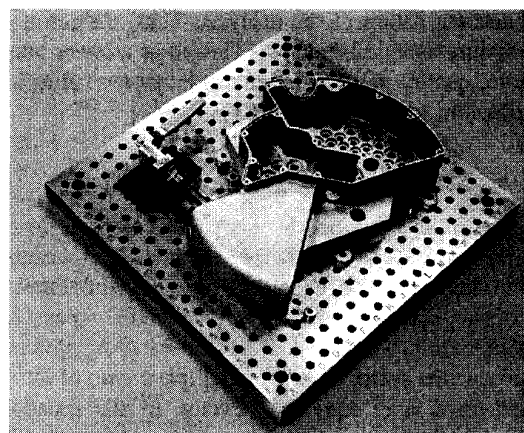
(c)

Fig. 13. A complex example. (a) A near-net-shape part that requires final machining operations. (b) The optimal fixture design returned by the algorithm. This example suggests a number of possible extensions to the current algorithm; see the text for details. (c) The part in the fixture.

returned fixture designs stems primarily from the part's large size relative to the fixture grid. If the grid were coarse, fewer designs would be produced, thus requiring less execution time.

TABLE I

| Grid Step	Candidate Fixtures	Final Fixtures	Run Time [hr:min:sec]
8	0	0	0:02:02
4	55	14	0:05:31
2	2,277	626	0:41:15
1	19,663	5,770	3:21:06

Fig. 14. Fixturing a hinge plate. (a) The hinge plate and the geometric access constraint required to add the second hinge plate. Because the geometric access constraint covers the entire right side of the hinge, no form-closure fixture is available. (b) A fixture that solves the problem by exploiting contact friction.

Our implementation accepts a grid-step parameter that allows the user to reduce the search space by effectively making the grid more coarse. For example, if grid-step = 2, then the algorithm only considers every other hole in the grid. Table I shows run time data for the electronic housing example of Fig. 13.

These data show that if a coarse grid resolution is explored, an initial set of fixture designs may be found in a matter of minutes, even for this complex example. As the search resolution is refined, the resulting fixture quality monotonically increases, since each successive computation at a finer resolution contains all of the fixtures designed at the previous resolution. This suggests a method of implementing the algorithm to support interactive design: First designs are generated on a coarse grid and reviewed, after which final designs are generated using a fine grid. This would allow designers to quickly obtain rough, "back of the envelope" fixture designs while considering the overall design picture, and then perform detailed fixture analysis later in the design phase. An even faster method of obtaining a single fixture design would be to run the enumeration using an arbitrary grid size, stopping as soon as an acceptable fixture is generated.

Note that regardless of the method used to generate a rough fixture design, the globally optimal design will not generally be "close to" the best design returned by the coarse analysis. Instead, the optimal design may correspond to a gross topological change in the fixture layout. This is an artifact of the discrete nature of the analysis. Thus, the result of the coarse fixture analysis should be viewed as a lower bound on the fixture quality that can be attained, rather than as a close approximation to the optimum fixture.

V. DISCUSSION AND FUTURE WORK

The algorithm described in this paper has been developed to design fixtures assembled from modular components; this algorithm may also be used to design dedicated fixtures. Modular fixturing components are somewhat more expensive than dedicated tooling, which makes dedicated tooling preferable in mass-production scenarios where multiple copies of the fixture are used for a large number of cycles. In this situation, the algorithm could be used to produce fixture designs which are fabricated with plain tooling plate. Since the major cost of modular fixturing components is in the precisely machined fixture plate, this scheme would produce cost-effective fixture designs. This approach has the additional advantage that prototype fixtures could be built from modular components for initial testing purposes.

The electronic housing example described in Section IV showed the result of applying the algorithm to a complex part,
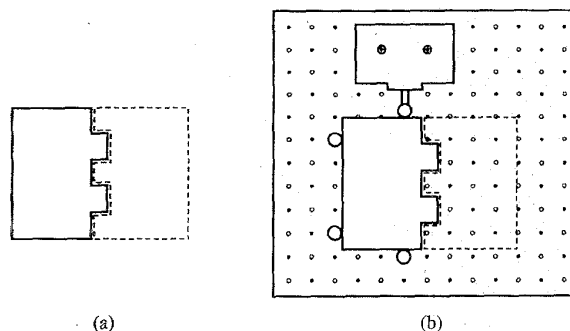
and motivated the use of variable-resolution search to provide both rapid response and detailed analysis. This example also illuminates a number of deficiencies in the current algorithm. These deficiencies are as follows.

- The algorithm does not design supports to hold the part above the plate. This is required to allow holes to be drilled through the bottom of the housing.
- The algorithm does not synthesize top clamp locations. Some machining operations produce forces in the $+z$ direction that tend to lift the part off the plate; these forces are only resisted by contact friction, which is not sufficient. In our housing example, lifting forces are generated by drilling and tapping operations.
- The algorithm does not allow curved edges in the part model. The curved edges of the housing are represented by a series of linear segments. This increases the combinatorics of the problem, reduces the accuracy of the estimated (x, y, θ) configuration, and causes some legitimate fixture designs to be missed due to false vertices in the part model.
- The algorithm does not synthesize redundant constraints. The fixtures designed by the algorithm are ideal three-point locating fixtures; these fixtures will provide unique and repeatable positioning of the part in the fixture. However, some situations will require additional contacts to adequately support the part. For example, the prototype version of the electronics housing was fabricated by machining it from a solid block of metal; this required substantial redundant support to prevent the thin walls of the part from chattering during machining operations.
- Some classes of parts fall outside the scope of the algorithm. For example, parts that are roughly cylinders lying on their side do not have the abundance of horizontal and vertical faces expected by this algorithm (e.g, aircraft landing gear). In these cases, fixturing strategies using V-blocks are common.
- The algorithm is limited to fixtures using four point contacts to constrain planar part motion. Certainly other fixture models can be imagined, for example including edge contacts or V-blocks. Catalogs from commercial suppliers typically include dozens of fixture modules

beyond the round locators and translating point clamps we consider. Often a combination of module types is required to fully constrain 3-D part motion.

Some of these problems are relatively easy to address. For example, synthesis of admissible support and top-clamp positions is straightforward for parts that have horizontal top and bottom surfaces, such as the electronics housing. In this sense, this algorithm comprises an essential part of a larger algorithm that synthesizes 3-D fixture designs for prismatic parts. Extending the algorithm to include curved edges requires implementation of the required ev–ev contact and force-locus equations for curved edges; the rest of the algorithm remains unchanged. However, substantial work is required to extend the algorithm to automatically synthesize redundant supports or support a broader class of modules. We hope to address all of these issues in future work.

Additional future work should focus on quality metrics. The quality metric presented in this paper is meant to provide an example of the kind of discrimination that is possible using the algorithm; this metric could be improved substantially. For example, the fixture in Fig. 2(a) was selected as optimal by our default quality metric. However, if the part is relatively soft, then it may pop out of the fixture due to deformation of the small edge contact at the upper right. In this case, the fixture shown in Fig. 2(b) is the better choice; this fixture was rated second-best by our default quality metric. One could imagine enhanced quality metrics that automatically perform a detailed analysis of part deformations or tolerances, or even heat dissipation in welding scenarios. Since the algorithm produces a list of all of the available fixture designs, the optimal fixture will be found relative to any metric that is provided.

Another possible extension of the algorithm would be to consider fixtures that exploit contact friction. The form-closure fixtures designed by the current algorithm provide the strongest possible constraint—part motion can only occur through deformation. On the other hand, sometimes this condition is too strong. For example, consider the problem of fixturing one plate of a common hinge, for the purpose of assembling the hinge. In order to allow the second hinge plate to be added to the assembly, one entire side of the hinge must be declared a geometric access constraint (see Fig. 14). This constraint is so large that no form-closure fixture is available. However, a variety of reasonable fixture designs may be applied to this task; Fig. 14(b) shows an example. This fixture relies on contact friction to prevent the part from moving toward the right. Extending our algorithm to include contact friction is straightforward. The primary change occurs in the form-closure analysis, where our current analysis of contact normals is replaced by an analysis of contact friction cones; see [38]–[40] for details. In addition, the enumeration procedure must be modified, since the part may be constrained with as few as two fixels when friction is present.

Other extensions could focus on expanding the set of available planar fixture designs. For example, our algorithm avoids contacts with part vertices. We chose this restriction for reasons of practicality, since vertices are more susceptible to deformation and are often less precisely located than part edges. However, in some cases vertex contacts would be practical, and extending the algorithm to include vertex contacts would increase the size of the solution space and possibly lead to better fixture designs for some problems. This could be accomplished by extending the enumeration to consider vertex contacts with fixture elements, invoking the necessary ve–ve and ve–ev c-space equations (some of which may be found in [39]), and including the diagonal edges of the zig-zag locus in the form-closure clamp placement analysis.

APPENDIX A
C-SPACE ANALYSIS

This appendix presents the equations required to determine the (x, y, θ) configuration of a polygon in contact with three point fixels. Each polygon edge makes an edge/vertex (ev) contact with a fixel; the possible placements of the polygon correspond to those placements where all three ev contacts are satisfied. This appendix provides a brief summary of topics developed in detail in [39].

The set of configurations that maintain a single ev contact has two degrees of freedom; these correspond to rotating the polygon about the fixel, or sliding the polygon laterally while maintaining the contact. Neglecting the edge endpoints and assuming a point fixel, this set corresponds to an unbounded surface that forms a helicoid in the (x, y, θ) configuration space. Because the polygon edge is not infinitely long, there are local applicability constraints that delimit the reachable subset of this infinite surface. Including these applicability constraints is straightforward, as described in [39]. We omit this complication here for brevity, but all of these details are included in our implementation.

The intersection of two such helicoidal surfaces forms a one-dimensional locus of points, which delineates the set of configurations that satisfy both of the corresponding ev contacts. This locus of configurations may occur in two forms, depending on the geometry of the corresponding polygon edges. If the polygon edges are parallel, then the surfaces will intersect in the (x, y, θ) space along lines parallel to the xy-plane, provided that the distance between the lines containing the edges is not larger than the inter-vertex distance. These lines describe the translational motion that is possible while maintaining both contacts. If the polygon edges are not parallel, then the surfaces will intersect along a curve that is a function of θ, describing the rotational motion that is possible while maintaining both contacts. We will refer to these as the parallel and nonparallel cases, respectively, and refer to the one-dimensional locus as an *ev–ev edge*.

In the general case, the intersection of three ev c-surfaces will define a finite collection of points. These points will also correspond to the intersection of the ev–ev edges corresponding to each pair chosen from the three surfaces. Thus we can find these intersection points by first forming two of the three ev–ev edges defined by the three c-surfaces, and then finding the points where the ev–ev edges intersect. In the remainder of this appendix, we will review our method of representing ev–ev edges and of computing the intersection points of two ev–ev edges.
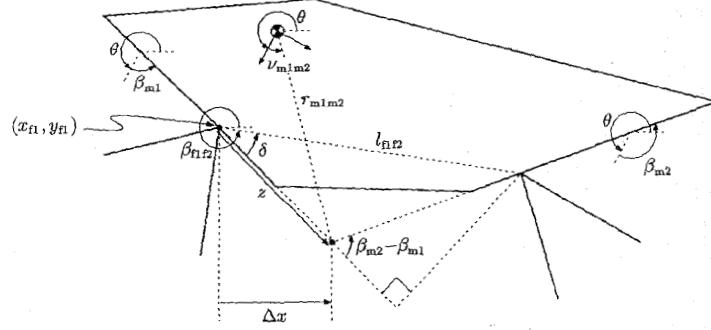
$$x = x_{f1} + \Delta x - r_{m1m2}c_{(\theta+\nu_{m1m2})}$$

$$x = x_{f1} + c_{(\theta+\beta_{m1})}z - r_{m1m2}c_{(\theta+\nu_{m1m2})}$$

$$x = x_{f1} + c_{(\theta+\beta_{m1})}\left[l_{f1f2}c_\delta - \frac{l_{f1f2}s_\delta}{\tan(\beta_{m2}-\beta_{m1})}\right] - r_{m1m2}c_{(\theta+\nu_{m1m2})}$$

$$x \approx x_{f1} + c_{(\theta+\beta_{m1})}\left[l_{f1f2}c_{(\beta_{f1f2}-\theta-\beta_{m1})} - \frac{l_{f1f2}s_{(\beta_{f1f2}-\theta-\beta_{m1})}}{\tan(\beta_{m2}-\beta_{m1})}\right] - r_{m1m2}c_{(\theta+\nu_{m1m2})}$$

$$f_x(\theta) \leftarrow k_{0_x} + k_{1_x}s_\theta + k_{2_x}c_\theta + k_{3_x}c_\theta s_\theta + k_{4_x}s_\theta^2$$

where

$$k_{0_x} = x_{f1} + l_{f1f2}c_{\beta_{m1}}\left[c_{(\beta_{m1}-\beta_{f1f2})} + \frac{s_{(\beta_{m1}-\beta_{f1f2})}}{\tan(\beta_{m2}-\beta_{m1})}\right]$$

$$k_{1_x} = y_{m1m2}$$

$$k_{2_x} = -x_{m1m2}$$

$$k_{3_x} = l_{f1f2}\left[-s_{(2\beta_{m1}-\beta_{f1f2})} + \frac{c_{(2\beta_{m1}-\beta_{f1f2})}}{\tan(\beta_{m2}-\beta_{m1})}\right]$$

$$k_{4_x} = -l_{f1f2}\left[c_{(2\beta_{m1}-\beta_{f1f2})} + \frac{s_{(2\beta_{m1}-\beta_{f1f2})}}{\tan(\beta_{m2}-\beta_{m1})}\right]$$

$$f_y(\theta) \leftarrow k_{0_y} + k_{1_y}s_\theta + k_{2_y}c_\theta + k_{3_y}c_\theta s_\theta + k_{4_y}s_\theta^2$$

where

$$k_{0_y} = y_{f1} + l_{f1f2}s_{\beta_{m1}}\left[c_{(\beta_{m1}-\beta_{f1f2})} + \frac{s_{(\beta_{m1}-\beta_{f1f2})}}{\tan(\beta_{m2}-\beta_{m1})}\right]$$

$$k_{1_y} = -x_{m1m2}$$

$$k_{2_y} = -y_{m1m2}$$

$$k_{3_y} = l_{f1f2}\left[c_{(2\beta_{m1}-\beta_{f1f2})} + \frac{s_{(2\beta_{m1}-\beta_{f1f2})}}{\tan(\beta_{m2}-\beta_{m1})}\right]$$

$$k_{4_y} = l_{f1f2}\left[-s_{(2\beta_{m1}-\beta_{f1f2})} + \frac{c_{(2\beta_{m1}-\beta_{f1f2})}}{\tan(\beta_{m2}-\beta_{m1})}\right]$$

Fig. 15. The functions $f_x(\theta)$ and $f_y(\theta)$ describing configuration obstacle edges that correspond to ev–ev contacts. The terms r_{m1m2}, ν_{m1m2}, β_{m1}, β_{m2}, x_{f1}, y_{f1}, l_{f1f2}, and β_{f1f2} are constants that depend on the shape of the grown edges e_1, e_2 and the positions of the fixels v_1, v_2. The first few lines show initial derivation steps for $f_x(\theta)$.

A. Representing EV–EV Edges

Recall that ev–ev edges may correspond to parallel or nonparallel cases. In the parallel case, the ev–ev edge corresponds to lines that are parallel to the xy-plane in the (x, y, θ) space. These may be represented by linear equations in x and y, with an associated θ value that describes the elevation of the line above the xy-plane. The derivation of these parameters is straightforward. In the nonparallel case, the ev–ev edge corresponds to a curve that is a function of θ; we represent this curve by two scalar functions: $f_x(\theta)$ and $f_y(\theta)$, which correspond to the x- and y-components of
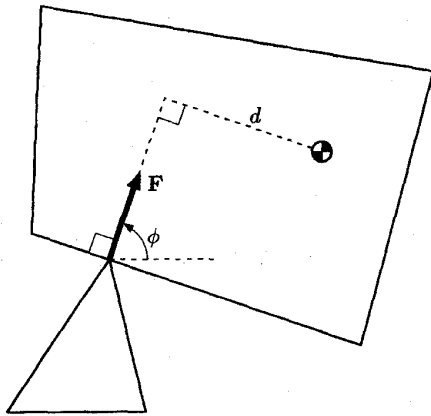
Fig. 16. Parameters used in mapping a contact normal ray **F** to its corresponding point on the force sphere. The parameter d is a signed quantity, and assumes a negative value when the reference point is on the opposite side of the ray **F**.

the curve, respectively. Fig. 15 shows the derivation of these functions.

B. Intersecting EV–EV Edges

Given a pair of ev–ev edges, there are three possible cases.

- **Parallel/Parallel.** In this case both ev–ev edges correspond to the parallel case, so all three polygon edges must be parallel. Thus there are either zero or an infinite number of solutions. In either case, we return no intersection points.
- **Parallel/Nonparallel.** The parallel-case ev–ev edge is defined by a collection of lines that lie on constant-θ planes. For each such line, we compute $x = f_x(\theta_{\text{line}})$ and $y = f_y(\theta_{\text{line}})$ and then test to see whether the resulting (x, y) point lies on the line. We return the set of all resulting intersection points; since there are at most two lines, no more than two intersection points may be returned.
- **Nonparallel/Nonparallel.** Given ev–ev edges A and B, the points where A and B intersect will be defined by the θ values which simultaneously satisfy the following equations:

$$f_{x_A}(\theta) = f_{x_B}(\theta) \tag{1}$$

$$f_{y_A}(\theta) = f_{y_B}(\theta). \tag{2}$$

These θ values are identified by finding the roots of (1) rearranged to $f_{x_A}(\theta) - f_{x_B}(\theta) = 0$, and then retaining all of the resulting θ values that also satisfy (2). There may be zero, one, or two consistent θ values; these are then supplied to the $f_{x_A}(\theta)$ and $f_{y_A}(\theta)$ functions to determine the corresponding x and y values. In certain degenerate situations it is possible to have an infinite number of solutions; this corresponds to the case where the two ev–ev edges are coincident. This case was observed by Farahat et al. [41].

Each of the above cases has an associated method for recovering the set of configuration-space points consistent with all three ev contacts. Typically this set will be comprised of up to two discrete points; in degenerate cases there may be an infinite set of consistent configurations. Since these degenerate situations do not constrain the object to a unique location, they are discarded by the fixture design algorithm.

APPENDIX B
FORCE-SPHERE ANALYSIS

This appendix presents the equations required to represent contact normals on the force sphere, and explains how to construct the locus of all possible contact normals for a given polygonal object. This is a brief summary of material previously presented in [38]–[40].

Erdmann developed a method for representing and analyzing planar forces in the $(F_x, F_y, \tau/\rho)$ space of planar forces [40]. The torque component of the space is scaled by the body radius of gyration ρ to provide analogous units for each axis of the space, and to facilitate vector analysis of the body dynamics. Erdmann showed how to analyze systems of forces by forming the convex hull of the forces in the force space, and in particular focused on dynamic analysis of bodies experiencing multiple frictional contacts. In Erdmann's method, a planar force is represented by a three-dimensional vector $\mathbf{F} = [F_x \quad F_y \quad \tau/\rho]^T$.

Brost and Mason observed that to characterize the set of possible motions, it is sufficient to consider only the direction of the force in $(F_x, F_y, \tau/\rho)$ space, neglecting its magnitude [38]. This led to two simplified representations of sets of planar forces: the dual-plane representation and the force sphere. Here we utilize the force-sphere representation, where forces in the $(F_x, F_y, \tau/\rho)$ space are projected onto the unit sphere centered at the origin.

In order to construct the set of form-closure fixtures that are possible for a given locator setup, there are two key constructions used by the algorithm: the convex hull of the contact normals for the locator setup, and the "zig-zag" locus of all possible contact normals for the given part.

For each construction, we first construct a set of force-space points that results from mapping particular contact normals onto the force sphere. This is accomplished by the following equation:

$$\mathbf{F} = \text{normalize} \begin{bmatrix} \cos(\phi) \\ \sin(\phi) \\ -\frac{d}{\rho} \end{bmatrix} \tag{3}$$

where ϕ and d are defined in Fig. 16, and the normalize operator scales the vector to unit magnitude. Recall that since we are not performing dynamic motion analysis, ρ may be any positive value.

Equation (3) gives us a method for mapping arbitrary contact normals onto the force sphere. To construct the convex hull of the contact normals for a locator setup, we first map each contact normal onto the force sphere, and then form oriented great-circle edges connecting all pairs of the three resulting points. Similarly, to construct the zig-zag locus of possible contact normals, we visit each polygon edge and map the contact normals corresponding to the edge endpoints onto the force sphere. Connecting the resulting pairs of force-sphere points will produce the desired vertical edges of the zig-zag locus.

ACKNOWLEDGMENT

REFERENCES

[1] C. Chang, "Computer-assisted fixture planning for machining processes," *Manufact. Rev.*, vol. 5, no. 1, pp. 15–28, Mar. 1992.

[2] B. Shirinzadeh, "Issues in the design of the reconfigurable fixture modules for robotic assembly," *J. Manufact. Syst.*, vol. 12, no. 1, pp. 1–14, 1993.

[3] F. Mason, "Modular tooling system zaps setup," *Amer. Machinist*, Mar. 1992.

[4] E. G. Hoffman, *Modular Fixturing*. Lake Geneva, WI: Manufact. Technol., 1987.

[5] F. Reuleaux, *The Kinematics of Machinery*. Macmillan, 1876. (Reprinted by New York: Dover, 1963.)

[6] Y. Zhuang, K. Y. Goldberg, and Y. Wong, "On the existence of modular fixtures," in *Proc. IEEE Int. Conf. Robotics and Automation*, May 1994, pp. 543–549. (Also USC Tech. Rep. IRIS-93-314.)

[7] E. N. Ohwovoriole, "Kinematics and friction in grasping by robotic hands," *Trans. ASME*, vol. 109, 1987.

[8] X. Markenscoff, L. Ni, and C. H. Papadimitriou, "The geometry of grasping," *Int. J. Robot. Res.*, vol. 9, no. 1, pp. 61–74, Feb. 1990.

[9] P. Somoff, "Uber gebiete von schraubengeschwindigkeiten eines festen korpers bei verschiedener zal von stutzflachen," *Zeitschrift fur Mathematik und Physik*, vol. 45, pp. 245–306, 1900.

[10] K. Lakshminarayana, "The mechanics of form closure," Amer. Soc. Mech. Eng., Tech. Rep. 78-DET-32, 1978.

[11] B. Mishra, "Workholding: Analysis and planning," In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Nov. 1991, pp. 53–57.

[12] H. Asada and A. B. By, "Kinematic analysis of workpart fixturing for flexible assembly with automatically reconfigurable fixtures," *IEEE Trans. Robot. Automat.*, vol. RA-1, no. 2, pp. 86–94, June 1985.

[13] M. R. Cutkosky, "Grasping and fine manipulation for automated manufacturing," Ph.D. dissertation, Dep. Mech. Eng., Carnegie Mellon Univ., Pittsburgh, PA, Jan. 1985.

[14] Z. Li and S. S. Sastry, "Task-oriented optimal grasping by multifingered robot hands," *IEEE Trans. Robot. Automat.*, vol. RA-4, no. 1, pp. 32–43, 1988.

[15] X. Markenscoff and C. H. Papadimitriou, "Optimum grip of a polygon," *Int. J. Robot. Res.*, vol. 8, no. 2, pp. 17–29, Apr. 1989.

[16] J. C. Trinkle, "A quantitative test for form closure grasps," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, July 1992, pp. 1670–1677.

[17] C. Ferrari and J. Canny, "Planning optimal grasps," in *Proc. IEEE Int. Conf. Robotics and Automation*, May 1992, pp. 2290–2295.

[18] D. Kirkpatrick, B. Mishra, and C. K. Yap, "Quantitative steinitz's theorems with applications to multifingered grasping," *Discrete Computat. Geometry*, vol. 7, pp. 295–318, 1992.

[19] J. J. Bausch and K. Youcef-Toumi, "Kinematic methods for automated fixture reconfiguration planning," in *Proc. IEEE Int. Conf. Robotics and Automation*, May 1990, pp. 1396–1401.

[20] P. J. Englert, "Principles for part setup and workholding in automated manufacturing," Ph.D. dissertation, Dep. Mech. Eng., Carnegie Mellon Univ., Pittsburgh, PA, Dec. 1987.

[21] H. Sakurai, "Automated setup planning and fixture design for machining," Ph.D. dissertation, Dep. Mech. Eng., Mass. Inst. Technol., Cambridge, Jan. 1990.

[22] S. H. Lee and M. R. Cutkosky, "Fixture planning with friction," *Trans. J. Eng. Ind.*, vol. 113, pp. 320–327, Aug. 1991.

[23] K. H. Kim, "A system for automated fixture planning with modular fixtures," Ph.D. dissertation, Robot. Inst., Carnegie Mellon Univ., Pittsburgh, PA, May 1993.

[24] M. V. Gandhi and B. S. Thompson, "Automated design of modular fixtures for flexible manufacturing systems," *J. Manufact. Syst.*, vol. 5, no. 4, pp. 243–252, 1986.

[25] Paul H. Cohen, "Automated fixture design," in *National Science Foundation Design and Manufacturing Systems Conf.*, Jan. 1991, pp. 405–413. (Published by the Soc. Manufact. Eng.)

[26] J. M. Schimmels and M. A. Peshkin, "Admittance matrix design for force-guided assembly," *IEEE Trans. Robot. Automat.*, vol. 8, no. 2, pp. 213–227, Apr. 1992.

[27] _____, "Force-assembly with friction," *IEEE Trans. Robot. Automat.*, vol. 10, no. 4, pp. 465–479, Aug. 1994.

[28] W. Boyes, Ed., *Handbook of Jig and Fixture Design*, 2nd ed. Soc. Manufact. Eng., 1989.

[29] B. Mishra, J. T. Schwartz, and M. Sharir, "On the existence and synthesis of multifinger positive grips," *Algorithmica*, vol. 2, no. 4, pp. 641–558, 1987.

[30] V. Nguyen, "Constructing force-closure grasps," *Int. J. Robot. Res.*, vol. 7, no. 3, pp. 3–16, June 1988.

[31] J. Ponce and B. Faverjon, "On computing three-finger force-closure grasps of polygonal objects," Beckman Inst., Univ. Illinois, Urbana-Champaign, Tech. Rep. UIUC-BI-AI-RCV-93-05, 1993.

[32] J. C. Trinkle and R. P. Paul, "Planning for dextrous manipulation with sliding contacts," *Int. J. Robot. Res.*, vol. 9, no. 3, pp. 24–48, June 1990.

[33] M. Mani, "Automatic design of workholding fixtures using kinematic constraint synthesis," Ph.D. dissertation, Dep. Mech. Eng., Northwestern Univ., Evanston, IL, Aug. 1988.

[34] Y-C. Chou, V. Chandru, and M. M. Barash, "A mathematical approach to automatic configuration of machining fixtures: Analysis and synthesis," *J. Eng. Ind.*, vol. 111, pp. 299–306, Nov. 1989.

[35] C. C. Hayes and P. K. Wright, "Automating process planning: Using feature interactions to guide search," *J. Manufact. Syst.*, vol. 8, no. 1, pp. 1–15, 1989.

[36] C. C. Hayes, "Machining planning: A model of an expert level planning process," Ph.D. dissertation, Robot. Inst., Carnegie Mellon Univ., Pittsburgh, PA, Dec. 1990.

[37] A. Wallack and J. Canny, "Planning for modular and hybrid fixtures," in *Proc. IEEE Int. Conf. Robotics and Automation*, May 1994, pp. 520–527.

[38] R. C. Brost and M. T. Mason, "Graphical analysis of planar rigid-body dynamics with multiple frictional contacts," in *Robotics Research: The Fifth International Symposium*, H. Miura and S. Arimoto, Eds. Cambridge, MA: MIT Press, 1990, pp. 293–300,

[39] R. C. Brost, "Analysis and planning of planar manipulation tasks," Ph.D. dissertation, School Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, Jan. 1991. (Also, Tech. Rep. CMU-CS-91-149, Carnegie Mellon Univ., Pittsburgh, PA.)

[40] M. Erdmann, "On a representation of friction in configuration space," *Int. J. Robot. Res.*, vol. 13, no. 3, pp. 240–271, June 1994.

[41] A. O. Farahat, P. F. Stiller, and J. C. Trinkle, "On the algebraic geometry of contact formation cells for systems of polygons," in review.

Randy C. Brost (S'85–M'89) received the B.S. degree in chemistry and computer science from the University of Denver, Boulder, in 1983, and received the Ph.D. degree in computer science from Carnegie Mellon University, Pittsburgh, PA, in 1991.

Since receiving the Ph.D. degree, he has worked as a Senior Member of the Technical Staff in the Intelligent Systems and Robotics Center at Sandia National Laboratories. His interests include robotic manipulation, automatic task planning and mechanics analysis, and robust computation. He has been developing new automatic planning and design algorithms for robotics and manufacturing applications since 1985.

Kenneth Y. Goldberg (S'84–M'90) received dual B.S. degrees in electrical engineering and management from the University of Pennsylvania, Philadelphia, in 1984. Working with M. Mason at the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, he completed the M.S. and Ph.D. degrees in 1987 and 1990, respectively. He also studied at Edinburgh University and the Technion.

From 1991 to 1995, he was on the faculty of the University of Southern California, Los Angeles, with a joint appointment in Computer Science and EE-Systems. In September 1995, he joined University of California at Berkeley's Department of Industrial Engineering and Operations Research. Currently, he and his group study algorithms for feeding, sorting, and fixturing industrial parts.

Dr. Goldberg serves on the Advisory Board of the IEEE Society of Robotics and Automation and is a National Science Foundation Presidential Faculty Fellow.