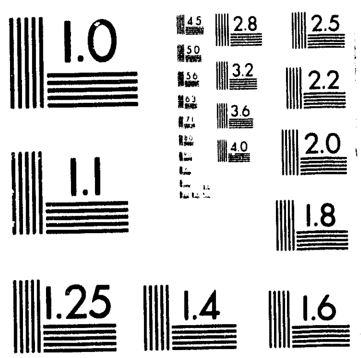

1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6

1 of 1

Submitted to the 1994 IEEE International Conference on Robotics and Automation.

A Complete Algorithm for Synthesizing Modular Fixtures for Polygonal Parts

Randy C. Brost*
Sandia National Laboratories
Albuquerque, NM 87185-5800

Kenneth Y. Goldberg†
University of Southern California
Los Angeles, CA 90089-0273

Commercially-available modular fixturing systems typically include a square lattice of tapped and bushed holes with precise spacing and an assortment of precision locating and clamping elements that can be rigidly attached to the lattice using dowel pins or expanding mandrels. Currently, human expertise is required to synthesize a suitable arrangement of these elements to hold a given part. Besides being time consuming, if the set of alternatives is not systematically explored, the designer may fail to find an acceptable fixture or may settle upon a suboptimal fixture.

We consider a class of modular fixtures that prevent a part from translating or rotating in the plane using four point contacts on the part's boundary. These fixtures are based on three round locators, each centered on a lattice point, and one translating clamp. We present an algorithm that accepts a polygonal part shape as input and synthesizes the set of all fixture designs that achieve form closure for the given part. The algorithm also allows the user to specify geometric access constraints on fixtures. If the part has n edges and its maximal diameter is d lattice units, the asymptotic running time of the algorithm is $O(n^5d^5)$. We have implemented the algorithm and present example fixtures that it has synthesized. This implementation includes a metric to rank fixtures based on their ability to resist applied forces. We believe this is the first fixture synthesis algorithm that is complete *in the sense that it is guaranteed to find an admissible fixture if one exists. Furthermore, the algorithm is guaranteed to find the* optimal *fixture, relative to any well-defined quality metric.*

1 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 such as the 3-2-1 rule and a number of studies, systematic algorithms for automatically synthesizing fixture designs based on CAD part models are still lacking [25].

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 advantages 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.

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 [12]. In this paper we present an algorithm for automatically synthesizing a class of modular fixtures. These fixtures prevent a part from translating and rotating in the plane by providing four contacts with the edges of the part's projected boundary. This class of fixtures includes three round *locators*, each centered on a lattice point, and one translating *clamp* that must be attached to the lattice via a pair of mounting 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. Since kinematic constraints may reliably resist arbitrary applied forces, we require fixtures to provide *form closure,* which is a kinematic constraint condition that prevents all motion. In addition to constraining the part, the fixture must not interfere with certain geometric regions, 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 enumerates all admissible fixtures and ranks them according to a user-definable scalar quality measure.

We believe this to be the first fixture synthesis algorithm that is *complete* in the sense that it guarantees

94AL85000
*Supported by US DOE contract DE-AC04-~~76DP00789~~.
†Supported by NSF awards No. IRI-9123747 and DDM-9215362, and by equipment grants from Adept Technology Inc, and Qu-Co, Inc.

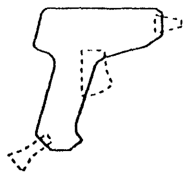


Figure 1: An example fixturing problem. The dashed regions must remain free of fixture components.

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. In a separate paper [27], we explore the existence or non-existence of fixtures for several classes of parts and fixture components.

Further, 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.

This paper is a condensed version of [5].

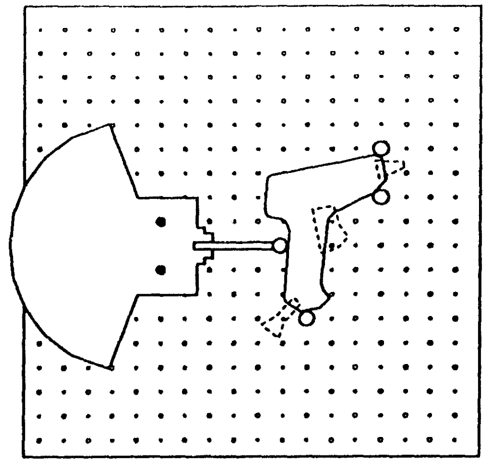
1.1 Example

Figure 1 shows an example where the part is one-half of the case of a commercially-available hot glue gun. We want a fixture to hold this part while assembling the gun. The dashes outline 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. In our experiments, we used a modular fixturing system from Qu-Co, Inc.

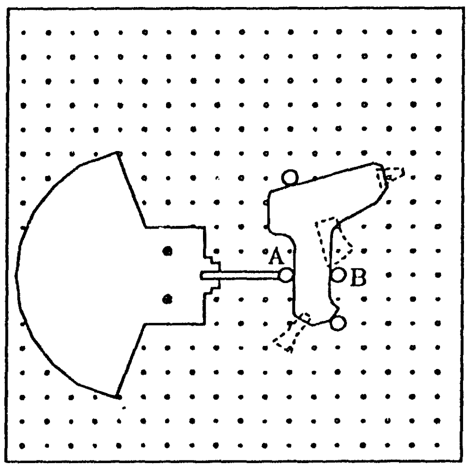
For this example, the algorithm returned 97 fixture designs, sorting them according to a quality metric which examines the maximum contact force required to resist an arbitrary unit applied force. Figure 2 shows two of the returned fixture designs. Note that both fixture designs provide form closure and obey the geometric access constraints.

If we consider a clockwise unit torque applied to the part, we see that the fixture in (a) is superior to the one in (b), where contacts A and B must exert very large contact reaction forces to resist the torque. Such considerations can be included in the metric used to rank fixtures. For example, the fixture design shown in (a) corresponds to the optimal design chosen by our default quality metric, while design (b) was rated as one of the worst designs. Section 3.6 describes this quality metric in detail. Figure 3 shows the part loaded into the assembled fixture.

In the remainder of this paper, we review related work and describe the algorithm in detail. We conclude with a discussion of directions for future work.



(a)



(b)

Figure 2: Two fixtures synthesized 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 synthesized 97 fixture designs in 129.4 seconds.

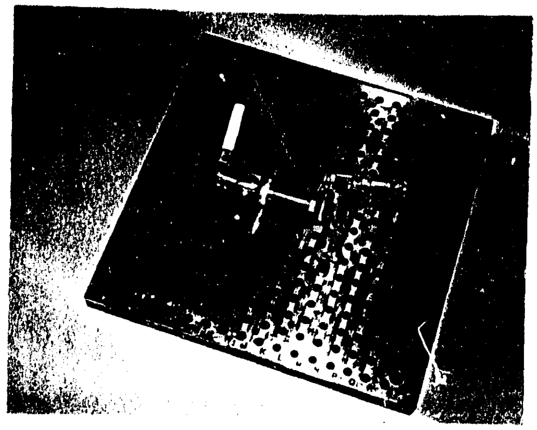


Figure 3: The assembled fixture of Figure 2(a).

2 Previous Work

There is a substantial literature on general fixturing methods including handbooks [3], kinematic methods for analyzing a given fixture, and heuristic methods for synthesizing fixtures. The century-old definition of *form closure* [23] captures the intuitive function of a fixture. A set of contacts provides form closure if infinitesimal part motion is completely constrained. Each contact provides a *wrench*: a force with a point of application. In the plane, a wrench can be represented as a vector in $\Re^3$, where the first two components represent the direction of force and the third component represents a moment about an arbitrary origin. A set of wrenches provides form closure if they positively span $\Re^3$. Results from linear algebra show that at least four wrenches are necessary for form closure. Recently, [18] showed that four wrenches are sufficient for any piecewise-smooth compact connected planar body, excluding surfaces of revolution.

In robotics, form closure has been studied in the context of grasping. For polyhedral parts in space, [15] showed that seven wrenches are necessary for form closure; [20; 18] showed that seven wrenches are sufficient. Nguyen [21] 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. For a part with n faces, his algorithm runs in time $O(n^c)$, where c is the number of independent regions. Such regions are useful because they allow for uncertainty in the part's pose prior to grasping. For a survey on robot grasping and quality measures of grasps, see [22].

Modular fixturing introduces additional constraints on the placement of contacts. For example, there have been a number of papers that address the problem of analyzing or synthesizing fixture designs using modular elements [1; 9; 8; 17; 7; 10; 24; 2; 16; 13; 25]. The synthesis methods in these papers are either based on heuristics or incomplete algorithms; none of the methods present a complete algorithm for synthesizing optimal fixture designs. However, these papers often include a detailed analysis of tolerances and part deformations, which we do not address here. We feel that these papers are excellent sources for improved quality metrics that include these important issues. See [11; 14] for surveys of the fixture design literature.

Mishra [19] addressed the problem of synthesizing modular fixtures from a computational perspective; he 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.

Recently, [26] reported an algorithm for synthesizing a class of modular fixtures with four round locators on a split lattice that can be closed like a vice. Their algorithm, like ours, takes the part shape as input and enumerates all combinations of locators that achieve form closure. Also, like ours, their algorithm sweeps edges to compute contact conditions and runs in polynomial time. The primary difference is that our model has a single lattice with one movable clamp and our implementation includes a force-based quality metric to rank fixtures.

3 The Algorithm

3.1 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 radius 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 at most one edge of the part. 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. Each fixture is specified by a data structure containing the following information:

- The (x, y) coordinates of the three locators.
- A pair of (x, y) coordinates describing the mounting position of the clamp on the plate.
- The (x, y, θ) coordinates of the part.

3.2 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 synthesized. This is accomplished by enumerating the set of possible locator setups, and then passing the result to a form-closure analysis that constructs all of the possible form-closure 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 slider, 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.

3.3 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 expanding the polygonal part boundary by the radius of the locators; fixturing the 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 edges 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 Figure 4.

3.4 Generating Candidate Fixtures

After transforming the problem so locators can be treated as points, 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 synthesize the set of all clamp positions that provide form closure for each locator setup. The following sections will explain each of these steps in detail.

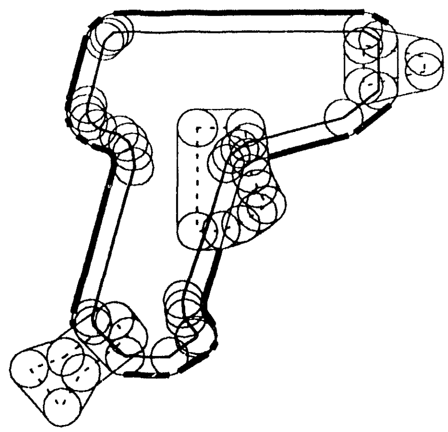


Figure 4: Transforming the input.

3.4.1 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. 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 the origin of the lattice. By translating and rotating e_a about the origin, the set of possible configurations for 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.

We now consider each of these second locator positions in turn, and identify all possible positions for the third locator. 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. 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 as shown in Figure 5.

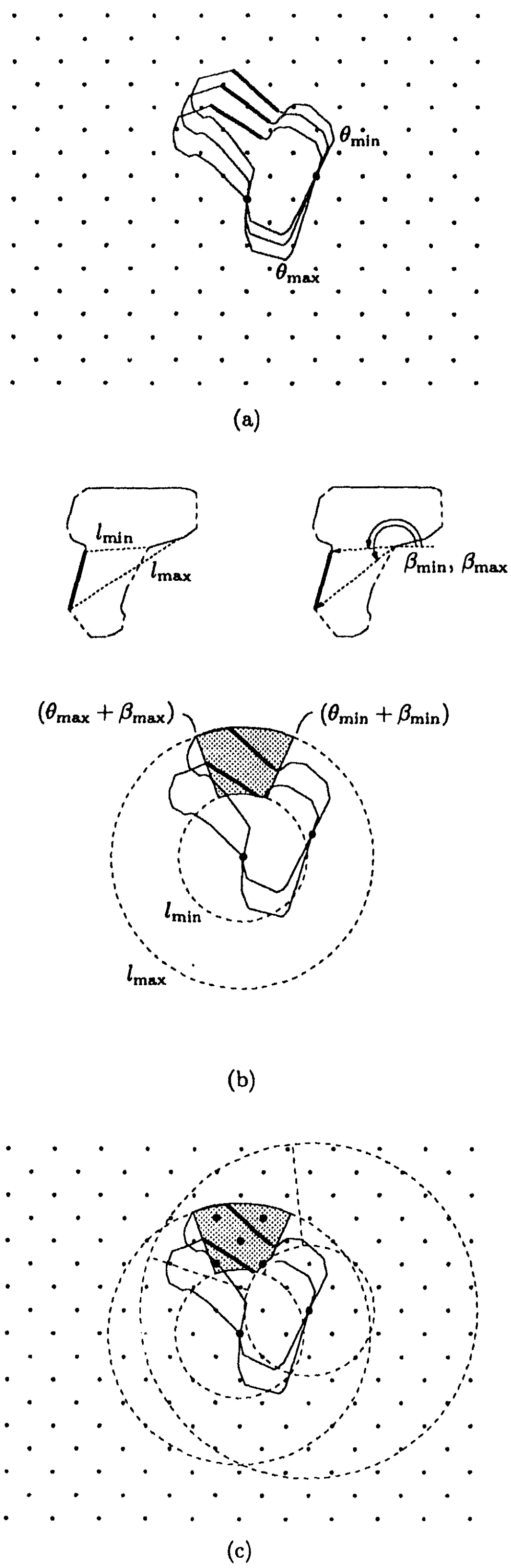


Figure 5: Identifying candidate locations for locator #3.

3.4.2 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. The required equations and associated solution methods are presented in detail in [4].

3.4.3 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.

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 [6]; see [4] for implementation details. This space represents both the direction and moment components of a line of force exerted in the plane.

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. Points a', b', and c' in Figure 6 show points on the force sphere corresponding to the contact normals of a typical locator setup.

A fixture design provides form closure exactly when the corresponding set of contact normals 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 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

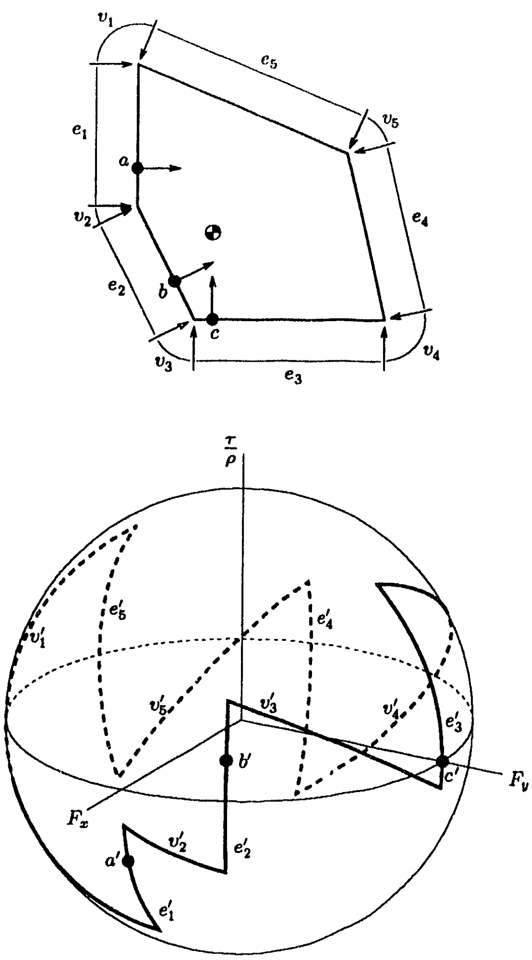


Figure 6: The "zig-zag" locus of all forces that may be exerted by contacting the part at a single point. This construction previously appeared in [6], and an analogous mathematical definition appeared in [20].

can be applied by a contact along the perimeter of the grown part. This set of forces is illustrated in Figure 6. Note that the set of all possible contact forces corresponds to a "zig-zag" locus of points that encircle the force sphere. Each point on the part boundary corresponds to a point on this locus. Edge contacts correspond to the vertical edges of the locus. By intersecting these vertical edges with the set of possible form-closure forces constructed previously, we can identify the set of all available 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 Figure 7.

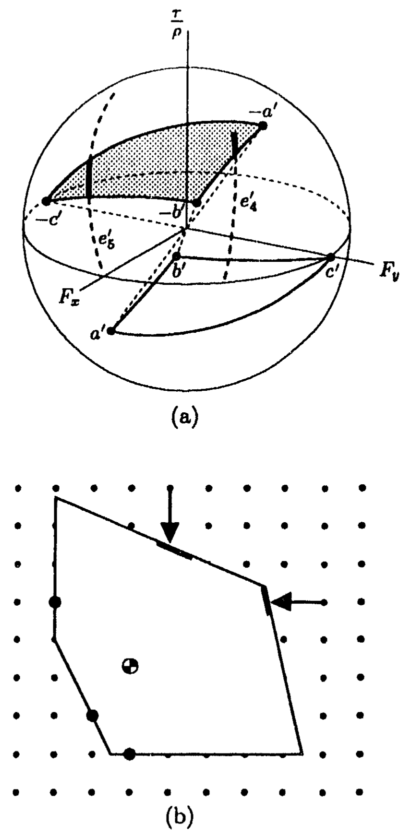


Figure 7: Constructing the set of form-closure clamp placements. (a) Force-sphere analysis. (b) Possible clamp placements.

3.5 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 slider 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 (this is a simple bounding-box check).

3.6 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 forces are undesirable because they may deform the part. The effect of fixture geometry on contact reaction force is illustrated in Fig-

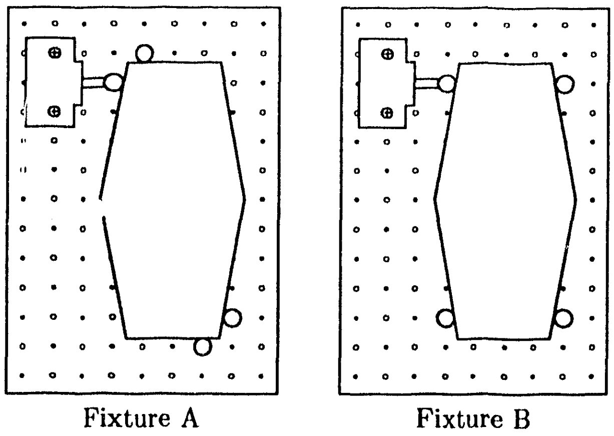


Figure 8: Two fixture designs. Which one is better? The answer depends on the forces that are expected.

ure 8. 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.

Our default quality metric allows the user to specify a variety of applied forces that are expected while the part is in the fixture. These forces are represented by a list of force-sphere regions with associated magnitudes; this allows the quality metric to simultaneously consider the effect of multiple 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 any of the expected applied forces; the quality score is the negative of this force magnitude.

4 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 Figure 1; in this problem the part has 28 edges, and the maximum diameter of the part is 136mm. The geometric access constraints are described by three polygons with a total of 20 edges. The fixture plate is a 18×18 array of alternating dowel/threaded holes, with a spacing of 19mm. The fixel radius is 12.7mm. The clamp and plunger have 23 and 4 edges, respectively. For this problem, the pure algorithm produced 23 grown part edges, 416 candidate fixtures, and 223 final admissible fixtures; this computation took 294.4 seconds.

Reviewing the output of the program suggests a number of simple heuristics for improving preformance 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 synthesis 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 1mm for the edge trim distance and minimum clamp clearance. This produced a significant performance improvement, reducing the computation time for the Figure 1 to 129.4 seconds. With these parameters, the program produced 17 grown part edges, 162 candidate fixtures, and 97 final fixtures.

5 Discussion and Future Work

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 $O(n^4d^5)$. Checking form closure and computing the quality metric for each fixture can be done in constant time. Checking for unwanted collisions can be accomplished in $O(n)$ time for each fixture, since the number of clamp edges is constant. Thus the algorithm runs in time $O(n^5d^5)$. We have not constructed a worst-case scenario; we believe that this bound can be tightened.

The current algorithm has a number of deficiencies:

- The algorithm does not consider the problem of loading the part into the fixture.
- The algorithm does not design supports to hold the part above the plate.
- 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.
- The algorithm does not allow curved edges in the part model. The curved edges of the glue gun were represented by a series of linear segments. This increases the combinatorics of the problem, reduces the accuracy of the estimated part position, and causes some legitimate fixtures to be missed due to pseudo-vertices in the part model.

We plan to address these issues and to experiment with different quality metrics. Our current algorithm is complete in the sense that if a fixture design exists for a given problem, the algorithm is guaranteed to find it. We would hope to retain this property while extending the algorithm to include the additional features described above. However, for some problems a fixture design may not exist, due to limitations of the fixture kit. We know of several such examples; see [27] for details. Another interesting area for future work would be to develop stronger characterizations of the class of parts that may be fixtured with a given fixture kit, and to develop planning algorithms for a richer fixture kit.

References

[1] H. Asada and A. B. By. Kinematic analysis of workpart fixturing for flexible assembly with automatically reconfigurable fixtures. *IEEE J. of Robotics and Automation*, RA-1(2), 1985.

[2] J. J. Bausch and K. Youcef-Toumi. Kinematic methods for automated fixture reconfiguration planning. In *IEEE Intl. Conf. on Robotics and Automation*, 1990.

[3] W. Boyes, editor. *Handbook of Jig and Fixture Design, 2nd Edition.* SME, 1989.

[4] R. C. Brost. *Analysis and Planning of Planar Manipulation Tasks.* PhD thesis, Carnegie Mellon School of Computer Science, 1991.

[5] R. C. Brost and K. Y. Goldberg. A complete algorithm for synthesizing modular fixtures for polygonal parts. Sandia Natl. Labs. SAND Report 93-2028, 1993.

[6] R. C. Brost and M. T. Mason. Graphical analysis of planar rigid-body dynamics with multiple frictional contacts. In H. Miura and S. Arimoto, editors, *Robotics Research: The Fifth Intl. Symposium*, MIT Press, 1990.

[7] Y-C. Chou, V. Chandru, and M. M. Barash. A mathematical approach to automatic configuration of machining fixtures: Analysis and synthesis. *J. of Engineering for Industry*, 111, 1989.

[8] P. J. Englert. *Principles for Part Setup and Workholding in Automated Manufacturing.* PhD thesis, Carnegie Mellon Dept. of Mech. Eng, 1987.

[9] M. V. Gandhi and B. S. Thompson. Automated design of modular fixtures for flexible manufacturing systems. *J. of Manufacturing Systems*, 5(4), 1986.

[10] C. C. Hayes and P. K. Wright. Automating process planning: Using feature interactions to guide search. *J. of Manufacturing Systems*, 8(1), 1989.

[11] F. B. Hazen and P. Wright. Workholding automation: Innovations in analysis, design, and planning. *Manufacturing Review*, 3(4), 1990.

[12] E. G. Hoffman. *Modular Fixturing.* Manufacturing Technology Press, Lake Geneva, Wisconsin, 1987.

[13] K. H. Kim. *A System for Automated Fixture Planning with Modular Fixtures.* PhD thesis, Carnegie Mellon Robotics Institute, 1993.

[14] L. S. King. Automatic fixture design for robotic assembly: Issues and current trends. In *NSF Design and Manufacturing Systems Conf.*, SME, 1992.

[15] K. Lakshminarayana. The mechanics of form closure. Technical Report 78-DET-32, ASME, 1978.

[16] S. H. Lee and M. R. Cutkosky. Fixture planning with friction. *Trans. of the J. of Engineering for Industry*, 113, 1991.

[17] M. Mani. *Automatic Design of Workholding Fixtures Using Kinematic Constraint Synthesis.* PhD thesis, Northwestern University Dept. of Mech. Eng, 1988.

[18] X. Markenscoff, L. Ni, and C. H. Papadimitriou. The geometry of grasping. *IJRR*, 9(1), 1990.

[19] B. Mishra. Workholding: Analysis and planning. In *Intl. Conf. on Intelligent Robots and Systems.* IEEE/RSJ, 1991.

[20] B. Mishra, J. T. Schwartz, and M. Sharir. On the existence and synthesis of multifinger positive grips. *Algorithmica*, 2(4), 1987.

[21] V. Nguyen. Constructing force-closure grasps. *IJRR*, 7(3), 1988.

[22] J. Pertin-Troccaz. Grasping: A state of the art. In O. Khatib, J. J. Craig, and T. Lozano-Pérez, editors, *The Robotics Review I.* MIT Press, 1989.

[23] F. Reuleaux. *The Kinematics of Machinery.* Macmillan, 1876. Republished by Dover, New York, 1963.

[24] H. Sakurai. *Automated Setup Planning and Fixture Design for Machining.* PhD thesis, MIT Dept. of Mech. Eng, January 1990.

[25] B. Shirinzadeh. Issues in the design of the reconfigurable fixture modules for robotic assembly. *J. of Manufacturing Systems*, 12(1), 1993.

[26] A. Wallack and J. Canny. Planning for modular and hybrid fixtures. In *IEEE Intl. Conf. on Robotics and Automation*, 1994. Submitted.

[27] Y. Zhuang, Y. Wong, K. Y. Goldberg, and R. C. Brost. Fixturability of parts. In *IEEE Intl. Conf. on Robotics and Automation*, 1994. Submitted.

DATE
FILMED

4/7/94

END