Modular Fixture Design for Generalized Polyhedra

Aaron S. Wallack [*] John F. Canny [†]
awallack@cognex.com jfc@cs.berkeley.edu
Computer Science Division,
University of California, Berkeley

Abstract

The term fixturing refers to the task of immobilizing a workpiece for the purpose of performing operations such as assembly and machining. As such, fixturing is of fundamental importance to industrial manufacturing. Fixtures can either be fabricated from scratch, or assembled from a toolkit of modular components, the latter approach is termed modular fixturing. In this paper, we present a complete algorithm to automatically design fixtures for the fixture vise toolkit. The algorithm enumerates fixture configurations, consisting of peg positions, workpiece pose, and jaw separation, for a given generalized polyhedral workpiece. A generalized polyhedral prismatic workpiece is defined to have a generalized polygonal silhouette, with a boundary composed of linear edges and circular arcs.

1 Introduction

The task of immobilizing a workpiece via mechanical devices, commonly called fixturing or workholding, is of fundamental importance to industrial manufacturing. Fixturing constitutes 10-20% of total manufacturing costs [2] because fixture costs are amortized over small batches (85% of all job shop batches consist of 50 or fewer workpieces). It is expensive and time consuming to fabricate fixtures from scratch, as compared to assembling fixtures from a modular toolkit of components; the latter approach, termed modular fixturing, can provide high precision, rigidity, durability, speed, and reconfigurability.

Since fixturing is of fundamental importance to manufacturing, its automation is greatly desired. In the past, fixture design has been more of a craft than a science. For this reason, fixtures and automated fixture design have been discussed at length in the mechanical engineering literature, manufacturing literature, and the robotics literature. Hazen and Wright's review [3] implied that most of the research in automated fixturing systems are based on expert systems. Gandhi and Thompson [2] attempted to bridge the gap between geometrical analysis and expert systems via a technique for analyzing force closure (the ability to resist arbitrary external forces and torques) within the framework of an expert system. The robotics community has also made significant contributions in terms of grasping, efficiently enumerating useful grasps, and grasp quality metrics; the reader is referred to [4, 8, 10] for an overview of these results. Mishra's [5] analysis of the toe clamp toolkit sparked renewed interest in the field of robotics.

Recently, Wallack and Canny [10, 11], Brost and Goldberg [1], and Overmars *et al.* [6] proposed modular fixturing toolkits which incorporated peg/hole devices (fixture tables) and a single degree of freedom, as well as complete fixture design algorithms. Wallack and Canny's [10, 11] and Brost and Goldberg's [1] algorithms share many similarities: both algorithms handle polygonal models, both rely on generate and test techniques. Wallack and Canny's fixture vise toolkit (Figure 1) is based on the Black and Decker Workmate, wherein modular fixture elements (pegs) are inserted into lattices of holes inlaid in fixture tables, where the fixture tables are mounted on jaws of a vise (Figure 1). Brost and Goldberg's translating clamp toolkit [1] involves three pegs and a translating clamp inserted into a single fixture table. Both toolkits share many advantages such as the property that closing the vise/tightening the clamp squeezes the workpiece into place; furthermore, when the fixture vise is turned upside down, it can function as a universal gripper.

The fixture vise provides two different types of configurations: Type I- where two pegs reside on each jaw of the vise, and Type II- where three pegs reside on one jaw of the vise, and one peg resides on the other. The translating clamp toolkit involves inserting three pegs and a translating clamp into a fixture table. We believe that the fixture vise toolkit is more general than

[*] Supported by Fannie and John Hertz Fellowship and NSF FD93-19412, currently at Cognex Corp., Natick, Massachusetts

[†] Supported by NSF FD93-19412

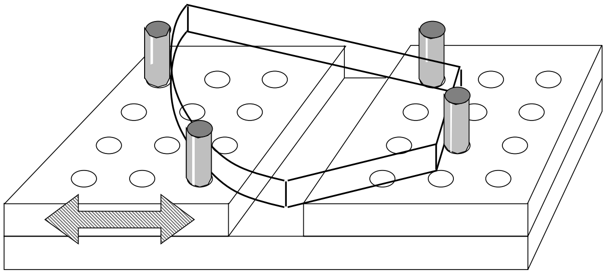


Figure 1: A fixture vise consists of two fixture table jaws capable of translating in x.

the the translating clamp toolkit because all translating clamp configurations correspond to Type II fixture vise configurations, where the jaw separation and peg on the right jaw act as the translating clamp. The two main differences between Wallack and Canny's fixture enumeration algorithm [10, 11] and Brost and Goldberg's fixture enumeration algorithm [1] is that Wallack and Canny computed the set of possible peg positions more conservatively, and Brost and Goldberg exploited the fact that for the translating clamp toolkit (and Type II fixture vise configurations), the workpiece's pose can be determined after choosing the first three peg positions. Thereby, Brost and Goldberg were able to exploit force closure constraints to prune the set of candidate fixture configurations.

In this report, we extend Wallack and Canny's fixture vise algorithm to handle generalized polygonal workpieces; a generalized polyhedral prismatic workpiece has the property that its projection onto the fixture plane is a generalized polygon, where a generalized polygon is defined as having a boundary composed of linear edges and circular arcs. The extensions we discuss are also applicable for extending Brost and Goldberg's fixture design algorithm [1] to handle generalize polyhedra.

1.1 Overview

In this paper, we describe the fixture vise design algorithm for generalized polyhedra, and highlight the differences from the original algorithm [10, 11]. The fixture design algorithm involves enumerating candidate quartets of features (model edges), enumerating candidate peg positions for each quartet of features, computing the poses achieving simultaneous contact for each combination of pegs features, and verifying force closure (the ability to resist arbitrary external forces and torques) for each simultaneous contact configuration.

We enumerate candidate peg positions using the annulus wedge strategy outlined in [1]; wedges of annuli liberally characterize regions swept over by one feature while maintaining contact between another feature and a point. In the previous algorithm ([10, 11]), we enumerated peg positions by parameterizing the object's pose in terms of the extended intersection of the first two linear features and first two peg positions; this parameterization exploited the property that the extended intersection of two linear features swept out a circle. Unfortunately, this property does not extend to circular features; consequently, we switched to a simpler, more liberal, method, involving ranges of orientations and distances between points on two features [1].

For each combination of features and pegs, we then compute the poses achieving simultaneous contact between circular and linear features and a set of points using a combination of classical algebraic methods and numerical techniques. In [10, 11], we solved for the simultaneous contact pose by formulating and intersecting pairwise constraints between contacts on the same vise jaw. For linear features, this approach reduces to solving at worst quartic polynomials. Extending this approach would have resulted in 32^{nd} degree polynomials. Instead, we used a different formulation which implicitly satisfies one of the four contact constraints; this approach reduces to solving 8^{th} order polynomials for Type II configurations, and 24^{th} order polynomials for Type I configurations.

1.2 Outline

In section two, we discuss the theoretical background and present an overview of the algorithm in section three. In section four, we compute the simultaneous contact poses. In section five, we present fixtures generated by our algorithm, and we conclude by highlighting the results and advantages of our technique.

2 Theoretical Background

2.1 Notation

- The operator $\oplus$ refers to the Minkowski sum: $A \oplus B = \{a+b | a \in A, b \in B\}$; we use the Minkowski sum to map $S^1 \times S^1 \rightarrow S^1$, $\mathbb{R}^1 \times \mathbb{R}^1 \rightarrow \mathbb{R}^1$

- O refers to a generalized polygon, a two dimensional region with a boundary composed of linear and circular features.

- $\vec{\mathcal{E}}$ refers to a quartet of *jaw-specified* linear or circular feature segments.

- $\vec{\mathcal{F}}$ refers a quartet of peg positions contacting the edge segments $\vec{\mathcal{E}}$. $\{\mathcal{F}_i\}$ refers to a set of a positions corresponding for the i^{th} peg.

- λ_x and λ_y refer to the column and row spacing on the modular jaws .

2.2 Planar Task

Since the workpieces are prismatic and the pegs are prismatic, we only need to consider the two dimensional projection (Figure 2). We assume that out of plane forces and torques are countered by other means. Still, the full three-dimensional problem is not intractable, as evidenced by Sudsang and Ponce [7] and Wagner and Goldberg [9].

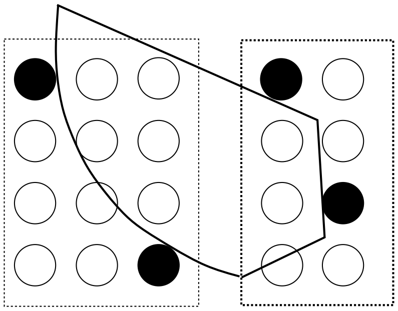


Figure 2: The two-dimensional view of the workpiece and the fixture vise in Figure 1.

2.3 Transforming Pegs to Points

As in [10, 11], the problem of achieving simultaneous contact with pegs is transformed to the problem of achieving simultaneous contact with points by appropriately growing the corresponding edges by the peg radii (Figure 3). For circular pegs, this reduction is valid, but for flatted pegs contacting linear features, this reduction is conservative. This point contact approximation provides a consistent framework for treating both types of pegs. The drawback of this assumption is that we require four pegs per configuration, even though three flatted pegs may provide force closure.

3 Algorithm

In this section, we describe algorithms and subroutines for enumerating peg configurations Figures 4-6 corresponding to a given quartet of edges. This technique bounds possible peg positions using geometrical constraints.

1. **Generate:**

   (a) Enumerate all quartets of jaw-specified feature segments $\{\vec{\mathcal{E}}, \vec{\mathcal{E}'}, \vec{\mathcal{E}''}, \ldots\}$ which can possibly achieve force closure with four point contacts (Figure 4).

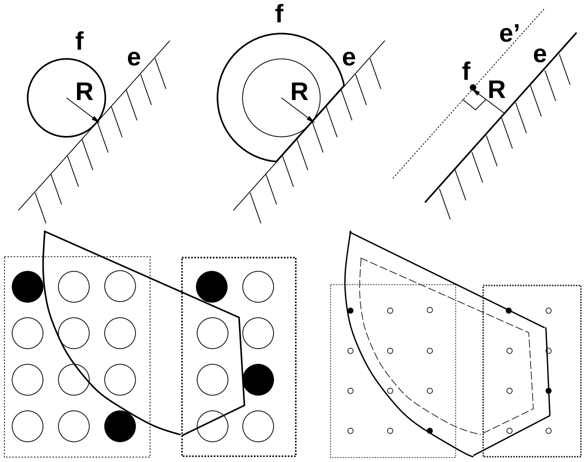


Figure 3: Cylindrical or flatted pegs can be considered point contacts by translating linear features outward by the peg radius, or changing the radius of circular features by the peg radius.

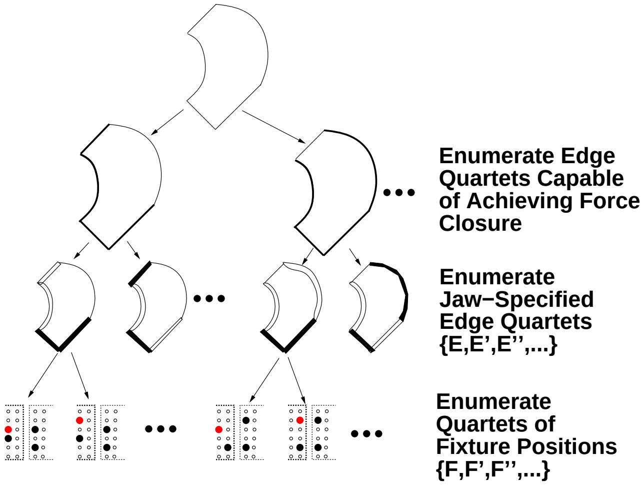

Enumerate Edge Quartets Capable of Achieving Force Closure
Enumerate Jaw−Specified Edge Quartets {E,E',E'',...}
Enumerate Quartets of Fixture Positions {F,F',F'',...}

Figure 4: The generate portion of the algorithm enumerates quartets of jaw specified feature segments, and for each edge quartet, enumerates quartets of fixture positions (represented by the filled-in circles) capable of achieving force closure.

   (b) For each feature segment quartet $\vec{\mathcal{E}}$, compute fixture configurations $\{\vec{\mathcal{F}}, \vec{\mathcal{F}'}, \vec{\mathcal{F}''}, \ldots\}$ providing simultaneous contact with $\vec{\mathcal{E}}$ (Figure 5). The fixture vise configurations $\{\vec{\mathcal{F}}, \vec{\mathcal{F}'}, \vec{\mathcal{F}''}, \ldots\}$ are enumerated as shown in Figure 6. We assume the first peg is placed at the origin, and then we enumerate all of the positions for the second peg. For each combination set of positions of the first and second pegs, we enumerate all of the posi-

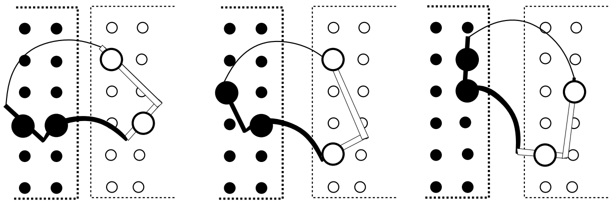


Figure 5: Different peg configurations $\vec{\mathcal{F}}$, $\vec{\mathcal{F}}'$, $\vec{\mathcal{F}}''$ simultaneously contacting edge segments $\vec{\mathcal{E}}$.

tions of the third and fourth pegs.

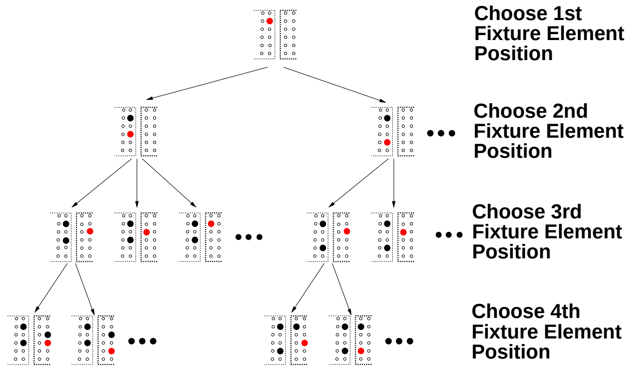

Choose 1st Fixture Element Position
Choose 2nd Fixture Element Position
Choose 3rd Fixture Element Position
Choose 4th Fixture Element Position

Figure 6: The peg configurations are enumerated by choosing the position of the second peg, then choosing the position of the third peg, etc. (filled-in circles represent the fixel positions).

2. **Test:**

   (a) Compute the workpiece poses and jaw separations achieving simultaneous contact between edge segments $\vec{\mathcal{E}}$ and peg positions $\vec{\mathcal{F}}$.

   (b) Verify force closure for each simultaneous contact pose.

3.1 Enumerating Peg Positions $\{\vec{\mathcal{F}}\}$

The first peg is placed at the origin on the left jaw. The positions of the second, third, and fourth pegs are bounded by geometrical constraints.

3.2 Valid Orientations: $\Theta_{L/R}$

The term $\Theta_{L/R}$ refers to the set of possible workpiece orientations allowing simultaneous contact between the left-jaw specified edges and pegs on the left jaw, and the right-jaw specified edges and pegs on the right jaw. For generalized polygonal workpieces, convex hulls of circular features corresponding to opposite jaws may overlap, and, in this case, all orientations satisfy left/right constraints. Also notice that $\Theta_{L/R}$ that can consist of two disjoint ranges; for example when $\Theta_{L/R} = \langle \frac{-\pi}{6}, \frac{7\pi}{6} \rangle \cap \langle \frac{5\pi}{6}, \frac{\pi}{6} \rangle$.

3.3 Annulus Wedges

Possible peg positions are characterized by a range of distances and ranges of orientations, which bound the possible vectors between points on the two features. Annulus wedges involve computing extremal orientations and distances. For linear features, the extremal orientations and distances correspond to extremal points on the features (the minimum distance may correspond to a vertex/non-vertex pair). For circular features, extremal orientations and distances may correspond to tangent points on the circular features. The the range can correspond to two distinct annulus wedges since $\Theta_{L/R}$ can correspond to two disjoint ranges.

3.4 Second Peg Positions $\{\mathcal{F}_2\}$

The second peg $\{\mathcal{F}_2\}$ corresponds to a lattice point consistent with the annulus wedge constraints. The orientation of the vector between $\mathcal{F}_1$ to $\mathcal{F}_2$ must lie within the Minkowski sum of $\Theta_{L/R}$ and the orientation ranges between points on $\mathcal{E}_1$ and $\mathcal{E}_2$ (Figure 7).

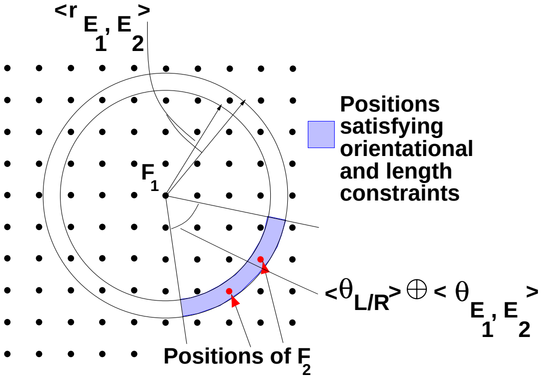


Figure 7: Possible peg positions $\{\mathcal{F}_2\}$ highlighted in grey are consistent with annulus wedge constraints where the annulus wedge is defined by $(\langle r_{\mathcal{E}_1,\mathcal{E}_2} \rangle, \langle \theta_{L/R} \rangle \oplus \langle \theta_{\mathcal{E}_1,\mathcal{E}_2} \rangle)$.

3.5 Third and Fourth Peg Positions for Type II Configurations $\{(\mathcal{F}_3, \mathcal{F}_4)\}$

For Type II configurations , since the first three pegs lie on the left vise jaw, $\mathcal{F}_3$ is constrained to satisfy annulus wedge constraints with respect to $\mathcal{F}_1$ and $\mathcal{F}_2$ (Figure 9): $\mathcal{F}_3$ must lie within the intersection of the two respective annuli wedges (Figure 8). We can then compute the workpiece's pose given the first three contacts, and given the pose, enumerate the fourth peg positions $\{\mathcal{F}_4\}$ on rows overlapping the transformed fourth edge.

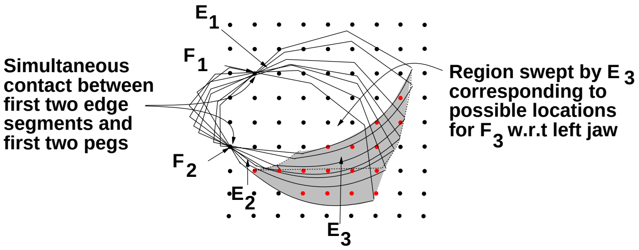

E1
F1
Simultaneous contact between first two edge segments and first two pegs
Region swept by E 3 corresponding to possible locations for F 3 w.r.t left jaw
F2
E2
E3

Figure 8: The third peg must lie in the region swept by $\mathcal{E}_3$ while maintaining contact between $\mathcal{E}_1$, $\mathcal{E}_2$ and $\mathcal{F}_1$, $\mathcal{F}_2$; this region is defined with respect to the left jaw. For Type II configurations, since the third peg lies on the same jaw as the first two pegs, the third peg corresponds to a lattice point within the swept region.

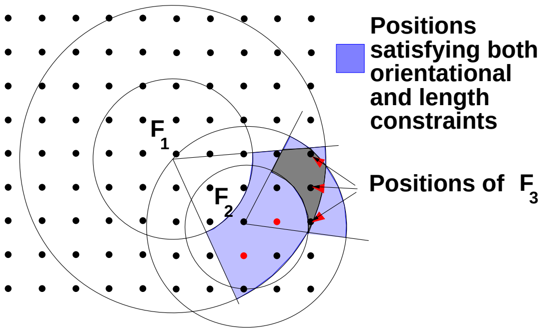

Positions satisfying both orientational and length constraints
F1
F2
Positions of F3

Figure 9: We bound the third peg positions $\mathcal{F}_3$ by intersecting two wedge annuli constraints: $\langle r_{\mathcal{E}_1,\mathcal{E}_3}\rangle$ and $\langle\theta_{L/R}\rangle \oplus \langle\theta_{\mathcal{E}_1,\mathcal{E}_3}\rangle$, $\langle r_{\mathcal{E}_2,\mathcal{E}_3}\rangle$ and $\langle\theta_{L/R}\rangle \oplus \langle\theta_{\mathcal{E}_2,\mathcal{E}_3}\rangle$.

3.5.1 Third and Fourth Peg Positions for Type I Configurations

For Type I configurations, enumerating peg positions is more difficult because the third and fourth pegs $(\mathcal{F}_3,\mathcal{F}_4)$ reside on the right vise jaw which may translate in x with respect to the first two pegs $(\mathcal{F}_1,\mathcal{F}_2)$ (Figure 10). Since the annulus wedge intersection regions for the third and fourth features were defined with respect to $\mathcal{F}_1$ and $\mathcal{F}_2$ these regions provide relative information about the possible x positions of the third and fourth fixture pegs. We compute pairs of possible peg positions $\{(\mathcal{F}_3,\mathcal{F}_4)\}$ as follows: Consider all pairs of rows corresponding to the two edge segments, $(\mathcal{R},\mathcal{R}^*)$, and the x regions included along those rows, $[X^{\mathcal{R}}_{\mathcal{E}_3}]$, $[X^{\mathcal{R}^*}_{\mathcal{E}_4}]$. Even though the x ranges $[X^{\mathcal{R}}_{\mathcal{E}_3}]$, $[X^{\mathcal{R}^*}_{\mathcal{E}_4}]$ were computed with respect to the left jaw, the differential between the x coordinate ranges $[X^{\mathcal{R}}_{\mathcal{E}_3}] \oplus (-[X^{\mathcal{R}^*}_{\mathcal{E}_4}])$ for $\mathcal{E}_3$ and $\mathcal{E}_4$ is invariant with respect to translation in x. We then enumerate pairs of peg positions $(\mathcal{F}_3,\mathcal{F}_4)$ for each discrete difference $(k\lambda_x \in [X^{\mathcal{R}}_{\mathcal{E}_3}] \oplus (-[X^{\mathcal{R}^*}_{\mathcal{E}_4}])(k \in \mathbb{Z})$, Figure 11).

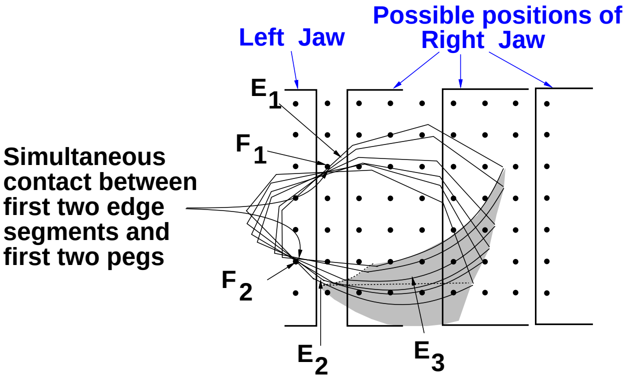

Left Jaw
Possible positions of Right Jaw
E1
F1
Simultaneous contact between first two edge segments and first two pegs
F2
E2
E3

Figure 10: In Type I cases, the right jaw is free to translate, so the swept region does not directly specifies the set of possible positions for the third and fourth pegs.

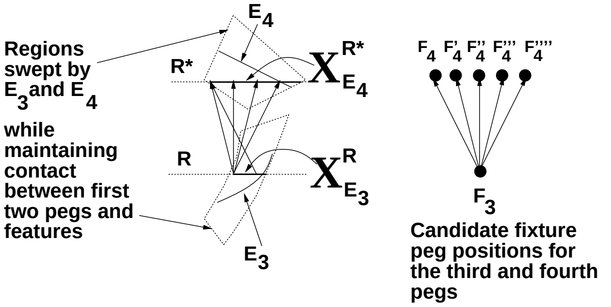

Regions swept by E3 and E4 while maintaining contact between first two pegs and features
E4
R*
X R* E4
R
X R E3
E3
F4 F'4 F''4 F'''4 F''''4
F3
Candidate fixture peg positions for the third and fourth pegs

Figure 11: We compute fixture positions $\{(\mathcal{F}_3,\mathcal{F}_4),(\mathcal{F}'_3,\mathcal{F}'_4),\ldots\}$ using the differential of the x coordinate ranges swept by the third and fourth edges $X^{\mathcal{R}}_{\mathcal{E}_3}, X^{\mathcal{R}}_{\mathcal{E}_4}$.

4 Computing Simultaneous Contact Poses

In this section, we present a method for computing simultaneous contact poses which transform model features onto pegs where one or two of the pegs may translate in x. We analyze this problem in terms of grown features contacting points. Our algebraic method uses parameterizations which implicitly maintain contact between the characteristic circular feature and the corresponding point. Since characterizing transformed points is easier than transforming features, we compute the transformation which maps the point set onto the model features, and then invert that transformation. We numerically solve for the orientation θ, and then solve for the translation parameters $(x,y)|_\theta$ using linear algebra.

4.1 Parameterizations

For Type I and Type II configurations, we rely on two different parameterizations which implicitly achieve contact between characteristic points and associated characteristic circular features (any circular feature can be the characteristic feature, the characteristic point is simply the point associated with that feature). In section 4.2, we discuss the pose parameterization used to solve for Type II cases which solves for the simultaneous contact pose between three features and three points. In section 4.4, we discuss the pose parameterization used to solve for Type I cases which must incorporate a degree of freedom to characterize the jaw separation.

4.2 (θ, ϕ) Parameterization: Type II

For Type II configurations, we use a parameterization (θ, ϕ) which corresponds to rotating the points by θ and then translating the points by $(R\cos(\phi), R\sin(\phi))$, where R refers to the radius of the characteristic circular feature. This parameterization implicitly achieves contact between the characteristic point and the characteristic circular feature assuming both the characteristic point, and characteristic circle, are initially centered at the origin.

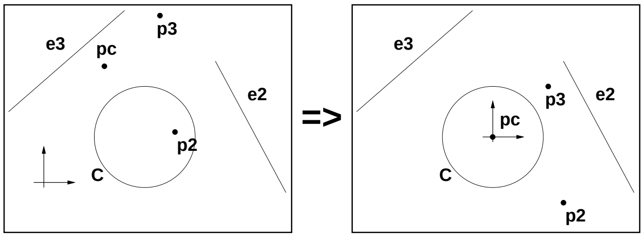


Figure 12: In order to utilize the parameterization, we first transform the points and features so that the center of the characteristic circle and the characteristic point both lie at the origin .

We then formulate the contact constraints algebraically. For Type II configurations, the simultaneous contact pose is found by solving the multivariate system corresponding to the other two contact constraints. Equation (1) characterizes the two-dimensional rigid transformation, $\mathsf{T}(\theta, \phi)$, in homogeneous coordinates. In order to arrive at algebraic expressions, we used the trigonometric substitutions $t = \tan(\frac{\theta}{2})$ and $u = \tan(\frac{\phi}{2})$.

$$\mathrm{T}(\theta, \phi) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & R\cos(\phi) \\ \sin(\theta) & \cos(\theta) & R\sin(\phi) \\ 0 & 0 & 1 \end{bmatrix} =$$

$$\mathrm{T}(t, u) = \begin{bmatrix} (1-t^2)(1+u^2) & -(2t)(1+u^2) & R(1+t^2)(1-u^2) \\ (2t)(1+u^2) & (1-t^2)(1+u^2) & R(1+t^2)(2u) \\ 0 & 0 & (1+t^2)(1+u^2) \end{bmatrix} \quad (1)$$

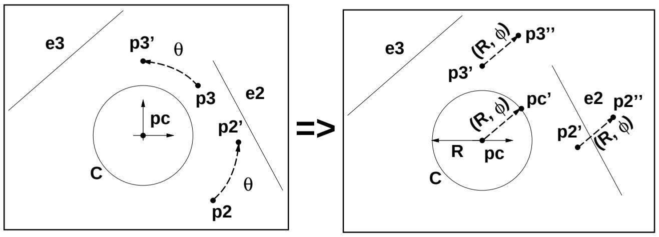


Figure 13: The Type II parameterization involves rotating the other 2 points around the origin by θ and then translating them by the vector expressed in polar coordinates as (R, ϕ). This parameterization implicitly achieves contact between the characteristic circular feature and associated characteristic points.

4.3 Contact Constraints

$h(t, u) = 0$ (equation (2)) expresses the constraint that a point is mapped onto a line and $g(t, u) = 0$ (equation (3)) expresses the constraint that a point is mapped onto a circle. (X, Y) refers to the point position, (a, b, c) characterize the line $a\mathtt{X} + b\mathtt{Y} = c$, and (xc, yc, r) characterize the circle $(xc - X)^2 + (yc - Y)^2 = r^2$. Fortunately the circular constraint expression is divisible by $(1+t^2)(1+u^2)$ yielding a biquadratic expression.

$$h(t,u) = \left([\mathsf{T}(t,u)] \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \right)^T \cdot \begin{bmatrix} a \\ b \\ -c \end{bmatrix} = 0 \quad (2)$$

$$K(t,u) = \left([\mathsf{T}(t,u)] \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \right) - \begin{bmatrix} xc \\ yc \\ (1+t^2)(1+u^2) \end{bmatrix}$$

$$g(t,u) = \frac{K(t,u)^T \cdot K(t,u) - r^2((1+t^2)(1+u^2))^2}{(1+t^2)(1+u^2)} = 0 \quad (3)$$

4.4 (θ, ϕ, σ) Parameterization: Type I

For Type I configurations, two of the points are free to translate in x. We use a similar parameterization (section 4.3) which incorporates the additional degree of freedom, the jaw separation σ (Figure 14). Since another (feature,point) pair resides on the same jaw as the characteristic feature, the contact constraint corresponds to the h, g expressions (section 4.3). The other two (feature,point) pairs which correspond to contacts on the opposite jaw must include a degree of freedom to characterize the jaw separation σ.

$h_\sigma(t, u) = 0$ (equation (4)) expresses the contact constraint corresponding to a point (X, Y) translated in x by σ and then transformed by (θ, ϕ) contacts a line. $g_\sigma(t, u) = 0$ (equation (5)) expresses the constraint that a point (X, Y) translated in x by σ

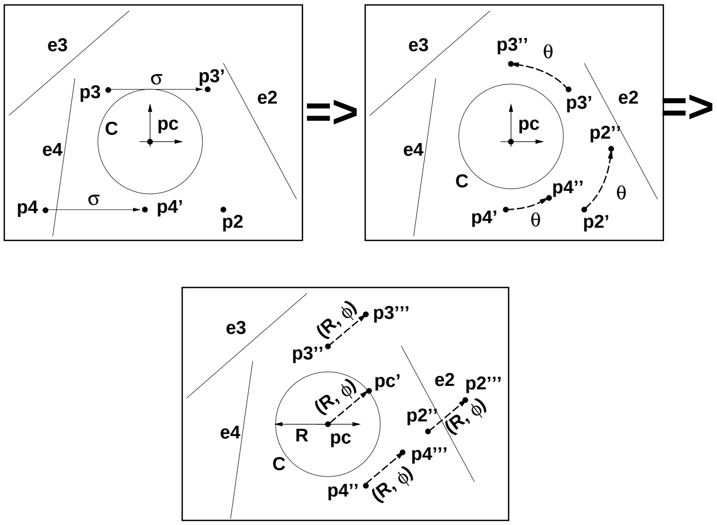


Figure 14: The Type I parameterization involves translating two of the points in x by σ and then rotating the 3 points around the origin by θ and then translating them by the vector expressed in polar coordinates as (R, ϕ).

and then transformed by (θ, ϕ) contacts the circle (xc, yc, r). Again the circular constraint expression is divisible by $(1+t^2)(1+u^2)$.

$$h_\sigma(t,u) = \left([\mathrm{T}(t,u)] \begin{bmatrix} X+\sigma \\ Y \\ 1 \end{bmatrix} \right)^T \cdot \begin{bmatrix} a \\ b \\ -c \end{bmatrix} = 0 \qquad (4)$$

$$K_\sigma = \left([\mathrm{T}(t,u)] \begin{bmatrix} X+\sigma \\ Y \\ 1 \end{bmatrix} \right) - \begin{bmatrix} xc \\ yc \\ (1+t^2)(1+u^2) \end{bmatrix}$$

$$g_\sigma(t,u) = \frac{K_\sigma(t,u)^T \cdot K_\sigma(t,u) - r^2((1+t^2)(1+u^2))^2}{(1+t^2)(1+u^2)} = 0 \qquad (5)$$

5 Results

In this section, we present example fixtures designed by the algorithm, and fixture counts for various workpieces. Table 1 lists the fixture counts and running times for various workpieces and λ_x, λ_y spacings, and c, the number of fixture vise columns. We include the specific case where $c = 1$ because it corresponds to using the fixture vise as a universal gripper with retractable pegs.

5.1 Example Fixtures

Fixtures generated by our algorithm are shown in Figures 15-18.

Workpiece	λ_y, λ_x	c	# (I/II)	Time (s)
thin4Handle	10,10	1	208 / 4	792
8Handle	10,10	1	36 / 0	356
small4Handle	10,10	1	14 / 0	395
small4Handle	10,10	10	88 / 123	9195
large4Handle	10,10	1	40 / 5	949
large4Handle	10,10	10	492 / 572	38968
oblongPart	10,10	1	26 / 0	112
oblongPart	10,10	10	280 / 198	4217
oblongPart	8,8	10	801 / 278	13896
widget	10,10	1	305 / 62	8644
widget	8,8	1	636 / 124	8896
flange	10,10	1	60 / 1	1742

Table 1: Fixture counts for various workpieces where c refers to the number of columns in the fixture vise

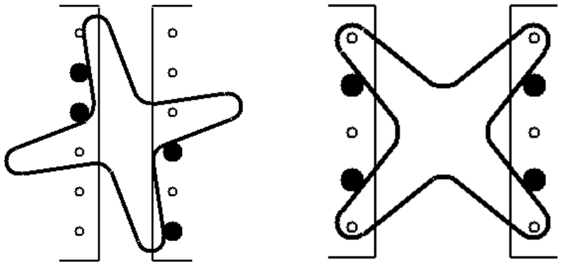
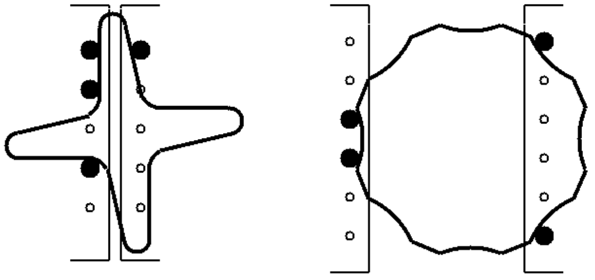


Figure 15: Two Type I fixtures for thin4Handle, a thin four sided handle

5.2 Degrees of Resultant Polynomials

We compute the simultaneous contact poses using numerical techniques and consequently, we are very interested in the complexity of the multivariate system (and corresponding univariate polynomials). The speed and precision of the numerical solver routines decrease as the polynomial degree, δ increases. Table 2 presents the polynomial degrees δ corresponding



(a) (b)

Figure 16: (a) a Type II fixture for thin4Handle (b) a Type I fixture for 8Handle, an eight sided handle

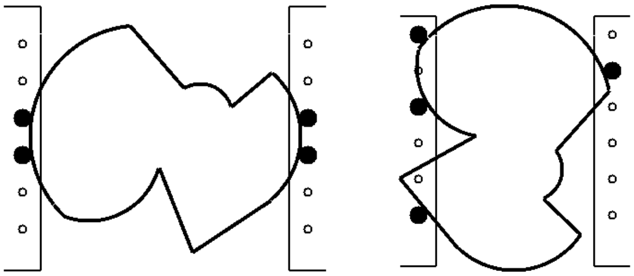


Figure 17: (a) a Type I fixture for widget (b) a Type II fixture for widget

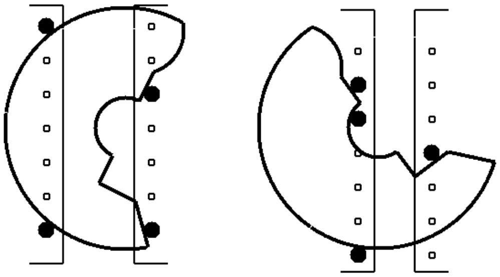


Figure 18: (a) a Type I fixture for flange (b) a Type II fixture for flange

to solving for θ for various scenarios;

Type	# Left Linear Features	# Left Circular Features	# Right Linear Features	# Right Circular Features	δ
I	2	0	2	0	4
I	1	1	2	0	12
I	0	2	2	0	12
I	1	1	1	1	24
I	1	1	0	2	24
I	0	2	0	2	24
II	3	0	1	0	2
II	3	0	0	1	2
II	2	1	1	0	4
II	2	1	0	1	4
II	1	2	1	0	6
II	1	2	0	1	6
II	0	3	1	0	8
II	0	3	0	1	8

Table 2: The maximum degree δ of the resultant expressions utilized for solving for each case

6 Conclusion

In this report, we extended the original fixture design algorithm [10, 11] to handle generalized polyhedral workpieces; we presented a complete, efficient algorithm for designing fixture vise configurations for generalized polyhedral prismatic workpieces. Since there are only a finite number of ways to fixture an arbitrary workpiece, fixtures can be enumerated in a generate and test manner. Simultaneous contact poses were computed algebraically using a novel parameterization which implicitly satisfies one of the three or four contact constraints, thereby decreasing the complexity of computing the simultaneous contact pose.

Acknowledgements

The authors wish to acknowledge: Dina Berkowitz for her assistance, Prof. Dinesh Manocha for valuable help in producing this paper, and Francesca Barrientos and Eric Paulos for helpful criticisms and suggestions on this report.

References

[1] Randy C. Brost and Ken Y. Goldberg. A complete algorithm for synthesizing modular fixtures for polygonal parts. In *International Conference on Robotics and Automation*, pages 535–542. IEEE, May 1994.

[2] M. V. Gandhi and B. S. Thompson. Automated design of modular fixtures for flexible manufacturing systems. *Journal of Manufacturing Systems*, 5(4):243–252, 1986.

[3] F. Brack Hazen and Paul Wright. Workholding automation: Innovations in analysis. *Manufacturing Review*, 3(4):224–237, December 1990.

[4] Brian Mirtich and John Canny. Easily computable optimum grasps in 2-d and 3-d. In *IEEE International Conference on Robotics and Automation*, San Diego, California, 1994.

[5] B. Mishra. Workholding: Anaylsis and planning. Technical Report No. 259, New York University, Courant Institute of Mathematical Sciences, November 1986.

[6] M. Overmars, A. Rao, O. Schwarzkopf, and C. Wentink. Immobilizing polygons against a wall. In *Symposium on Computational Geometry*, pages 29–38, Vancouver, BC, June 1995. ACM, ACM Press.

[7] Attawith Sudsang and Jean Ponce. New techniques for computing four-finger force-closure grasps of polyhedral objects. In *IEEE International Conference on Robotics and Automation*, pages 1355–1360, 1995.

[8] J. C. Trinkle. On the stability and instantaneous velocity of grasped frictionless objects. *IEEE Transactions on Robotics and Automation*, 8:560–572, 1992.

[9] Richard Wagner and Ken Goldberg. Strut based modular fixturing. In *IEEE International Symposium on Assembly and Task Planning*, 1995.

[10] Aaron Wallack. *Algorithms and Techniques for Manufacturing*. PhD thesis, University of California at Berkeley, 1995.

[11] Aaron Wallack and John Canny. Planning for modular and hybrid fixtures. *Algorithmica*, 1996. to appear.