

Immobilizing Polygons against a Wall

Mark Overmars* Anil Rao† Otfried Schwarzkopf‡
Chantal Wentink*

Abstract

A familiar task in industrial applications is grasping an object to constrain its motions. When the external forces and torques acting on the object are uncertain or varying, form-closure grasps are preferred; these are grasps that constrain all infinitesimal and finite motion of the object. Much of previous work on computing form-closures has involved achieving it with point-contacts; for a planar object, four point-contacts were proven to be necessary and sufficient. Inspired by the intuitive habit of supporting an object against something flat to immobilize it, in this paper we propose a new class of contacts called edge-contacts; these offer a straight-line support against which the object rests. Our first result is that almost any polygonal part can be constrained in form-closure with an edge-contact and two point-contacts.

A related problem is that of immobilizing an object with modular fixtures. These typically comprise of a regular lattice of holes on which the object is placed and a set of precision contacting elements or fixels whose locations are constrained by the grid. Given the input object modeled as an n -gon, the grid size, and the shapes of the fixels available, we present an algorithm that returns all valid modular fixtures, each accomplished by an edge-fixel, a locator-fixel and a clamp. The algorithm uses sophisticated data-structures to achieve output-sensitivity; it runs in $O(n(n+p)^{4/3+\epsilon} + K)$ time, where K is the number of valid solutions (fixture configurations) returned, and p is the object's perimeter in grid units.

*Department of Computer Science, Utrecht University, P.O.Box 80.089, 3508 TB Utrecht, the Netherlands, {markov, chantal}@cs.ruu.nl. This research was supported in part by ESPRIT Basic Research Action No. 6546 (project PROMotion) and the Dutch Organization for Scientific Research (NWO).

†This work was done when the author was with the department of Computer Science in Utrecht. The author is now with Qualcomm, Inc., 6455 Lusk Blvd., San Diego, CA 92121-2779. Email: arao@qualcomm.com

‡This work was done when the author was with the Department of Computer Science in Utrecht. He is now at the Department of Computer Science, Postech, San 31 Hyoja-dong Pohang, Kyungbuk 790-784, South Korea. Email: otfried@vision.postech.ac.kr

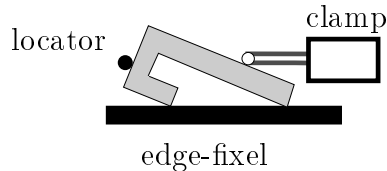


Figure 1: An example fixture with an edge-fixel, a locator and a clamp.

1 Introduction

Many manufacturing operations, such as machining, assembly, and inspection, require constraints on the motions of parts or subassemblies of parts [4, 6]. The concept of *form-closure* is over a century old [15] and refers to constraining, despite the application of an external wrench (force and moment), all motions of a rigid object (including infinitesimal motions) by a set of contacts on the object; any motion of an object in form-closure has to violate the rigidity of the contacts. Therefore, the problem becomes computing contact locations on a given part shape that achieve form-closure.

In this paper, we are interested in immobilizing planar objects and in particular, polygons. We refer to the set of contacts achieving form-closure as a *fixture*. We assume the contacts are frictionless; note that this is a conservative assumption since any fixture computed assuming zero friction also holds in presence of friction. By the *fixture model*, we imply the set of allowable contact types. The conceptually simplest model is that of point contacts. It has been known since Reuleaux that fixturing a planar object requires at least four frictionless contacts. Mishra, Schwartz, and Sharir [13] and Markenscoff, Ni and Papadimitriou [11] independently proved that four point-contacts are also sufficient.

While point-contacts are conceptually simple, they are not always easy to achieve in practice. The reason is that for form-closure, these point-contacts have to be capable of resisting arbitrary wrenches and therefore they have to be backed by bulky supports. This in turn implies difficulty of placing point-contacts at some points on the boundary of an object, in particular, at narrow concavities. Therefore it becomes important to look for other possible practical fixture models in order to reduce the number of point-contacts. In everyday life, we frequently lean an object against a flat surface, such as a table or a wall, to constrain its motions. In the planar world, the analog of a wall is a supporting line. In this paper we extend the fixture model to include *edge-contacts* which offer straight-lines of support. Notice that an edge-contact can touch the object only along its convex hull. The object simply rests against it; there is no reaching into concavities. See for an example Fig. 1.

How powerful are edge-contacts with respect to point-contacts? Although we cannot place point-contacts at convex vertices, it is possible to place edge-

contacts touching a pair of convex vertices (a pair of non-adjacent vertices of a non-convex polygonal object but adjacent in the convex hull). On the other hand, in general, an edge-contact cannot replace more than two point-contacts. To see this, consider a convex polygonal object; an edge-contact is equivalent to two point-contacts at the two ends of contact. The question remains whether an edge-contact can always replace two point-contacts; in other words, can form-closure on polygons be achieved with one edge- and two additional point-contacts? It is easy to show that rectangular objects cannot be immobilized this way; there is always a direction to slide the rectangle out. However, we answer the question in the affirmative for any polygonal object which has no edge parallel to a different edge of its convex hull. Our proof is constructive and produces a fixture in $O(n \log n)$ time. Objects that have edges parallel to convex hull edges, a condition that can be prechecked in $O(n \log n)$ time, may or may not be fixturable.

The second part of our work relates to *modular fixtures* which is a subject of considerable popular interest in the manufacturing industry for the past ten years or so [2, 3, 8, 9]. Basically, this involves a regular square grid of lattice holes accompanied by fixture elements (or *fixels* [5]) that are constrained by the grid; the object then rests against these fixels which constrain its motions. Custom-built fixtures being expensive, the major benefit of modular fixtures stems from their reconfigurability; it is often necessary to fixture an object only for short periods of time after which the same set of fixels can be used to fixture different objects. Another advantage of modular fixtures is their easy assembly and disassembly.

Since research in computing form-closures generally involved point-contacts, it is not surprising that most fixels were designed to achieve point-contact. The simplest fixel is the *locator* which is a circular object centered at a lattice hole. Since achieving contact with four circles (constrained to a grid) is in general impossible ([17] shows this even for three circles), it is clear that we need a fixel that takes care of the slack. Such a fixel is called a *clamp* which has a fixed portion, the clamp *body*, attached to a movable rod, the clamp *plunger*, that can translate between certain limits along a grid line. The end of the plunger is the clamp *tip* which makes contact with the object. A clamp can be configured so that the motion of the plunger is parallel to either one of the axes; it is termed *horizontal* or *vertical* accordingly. See Fig. 1 for an example of a clamp.

Much of the research in modular fixtures to date has been on fixture analysis with detailed mechanical studies on part deformations and tolerancing; the fixture synthesis algorithms are generally incomplete or heuristic in nature. Among exceptions are the works by Mishra [12] and Brost and Goldberg [5] who investigated fixturing polygons with three locators and one clamp (the 3L1C model). However, these are either limited to simple object shapes ([12]: rectilinear objects) or are algorithmically inefficient and not output-sensitive because they involve some kind of a brute-force search for fixture locations among tuples of edges ([5, 16]).

Wallack and Canny [16] consider an interesting model of modular fixtures which uses four locators and no clamps; instead, the slack is countered by mounting the part on a split horizontal lattice and allowing the one half to slide horizontally relative to the other. They also give an algorithm to compute all possible fixtures which runs in time proportional to the number of part configurations contacting four locators. Notice that this is not output sensitive in the actual number of fixtures output as many contacting configurations may indeed not be in form closure.

Brost and Goldberg [5] and Wallack and Canny [16] do not identify a class of fixturable objects while [12, 17] identify a subset of rectilinear objects as fixturable under the respective model considered.

Corresponding to the edge-contact discussed before and to enhance our fixture model, we add *edge-fixels* to our repertoire. An edge-fixel is simply a bar-like object of appropriate dimensions fixed to the lattice offering a straight-edge of support. We assume that an edge-fixel is at least as long as the longest edge in the convex hull of the object. As before, we consider replacing two of the point-fixels (locators/clamps) by an edge-fixel and we describe an algorithm that generates the (possibly empty) list of all valid form-closure fixtures generated by one edge-fixel, one locator, and one clamp. We term this the ELC model.

An important distinguishing characteristic of our algorithm compared to previous algorithms is that it is output sensitive; it runs in $O(n(n+p)^{4/3+\epsilon} + K)$ time for any positive constant ϵ , where K is the number of fixtures output which can be as high as $n(n+p)^2$. By feeding these solutions through a given quality metric or Boolean comparator, an optimal fixture may be computed. Optimization criteria may include some function of the distance between the points of contact, or a function of the directions of contact (to prevent wear-and-tear, it might be necessary to orient fixels as far as possible towards or away from a particular direction), *etc.* The generated solutions can be sorted according to any given metric. The fixture that is the best under one metric may be bad under a different one; therefore, it is usually not sufficient to generate just one valid fixture but instead the whole set.

After introducing some preliminaries in Section 2, we consider fixturing polygons with an edge- and two point-contacts in Section 3. The results of this section are equivalent to considering locators and clamps of negligible size and locatable anywhere on the object's boundary. Section 4 considers the case of modular fixtures: a finitely sized grid and fixels. Finally, we close with some remarks and suggestions for future work in Section 5. Since there is a page limitation, we cannot afford to present all proofs here; these will be presented in a forthcoming paper.

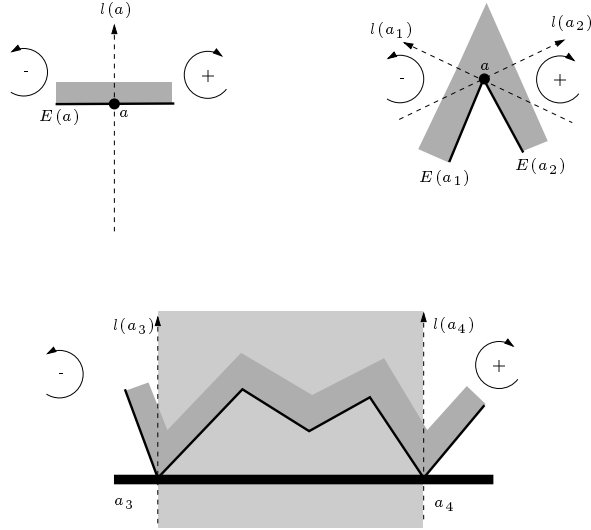


Figure 2: Motions possible under simple contacts.

2 Preliminaries and Notation

Recall that a fixture is said to provide *form-closure* if it precludes all (planar) motion, translations and rotations. Let us begin by examining what motions are ruled out by single point- or edge-contacts. From here onwards, we will consider rotations only with the understanding that translations in a direction are simply rotations about infinity along the perpendicular direction. Note that any infinitesimal motion of a polygon can be represented by a point in the plane, denoting the center of rotation (COR) of this motion. A point at infinity thus represents a infinitesimal translational motion.

Denote the input polygon by P and its boundary by ∂P ; n is the number of edges forming ∂P . Let the polygon edge containing a point a on its boundary be denoted by $E(a)$; the directed line perpendicular to $E(a)$ through a pointing into the interior of P is $l(a)$. We distinguish between the following cases (See Fig. 2).

Point-contact at interior of an edge This is the fundamental contact and the motions allowed by other types of contacts can be deduced by composing those allowed by elementary point-contacts.

Consider a point-contact at point a in the interior of edge $E(a)$ as shown. The allowed motions are defined by the line $l(a)$. If the object rotates (infinitesimally) in a clockwise (positive) direction with a point-contact at a , its COR will have to lie in the region to the right of (and including) $l(a)$. Furthermore, any point in this closed right-half-plane is a possible COR for a positive rotation. Similarly, the CORs for counter-clockwise or negative rotations lie in the closed half-plane to the left of $l(a)$. These are all and the

only constraints imposed by the point-contact at a . For future reference, we term these as *half-plane constraints* imposed by the point contact.

Point-contact at concave vertex This is shown in the upper-right of Fig. 2; a is the concave vertex. Imagine two points a_1, a_2 infinitesimally close to a along the two edges defining it. The motions allowed by a can be determined by intersecting the motions allowed by the point-on-edge pairs $(a_1, E(a_1))$ and $(a_2, E(a_2))$ which may be individually analyzed as above. The result is shown in the figure. These may be termed the *wedge constraints* defined by a .

Consider rotating a line l through a from $l(a_1)$ to $l(a_2)$ in the clockwise direction. The wedge constraints defined by a is the intersection of each of the half-plane constraints obtained along the sweep. Therefore, the motions allowed by a are a subset of the motions allowed by any one of these half-plane constraints. However, the event that concave vertex is on a grid line is a degenerate one and in practice this will rarely happen. So, in future, for ease of analysis, we choose to approximate the wedge constraint by a half-plane constraint defined by a particular line along this sweep.

Edge-contacts An edge-contact can contact the polygon along an edge e of the polygon or along two vertices a_3, a_4 (adjacent on the convex hull). The latter case is shown in the figure; the former case can be similarly analyzed considering a_3, a_4 to be the end-vertices of e . Consider lines $l(a_3), l(a_4)$ perpendicular to the edge-contact and superimpose the two half-plane constraints to get the result shown. The closed left half-plane allows negative rotations while that on the right allows positive rotations. The open infinite “slab” in the middle, shown shaded in the figure and disallowing all motions is denoted $slab(a_3, a_4)$ and is a crucial entity in future analysis. The constraint imposed will be called a *slab constraint*.

3 A Class of Fixturable Objects

In this section we prove that any polygon having no edge parallel to a different edge in its convex hull can be immobilized with one edge- and two point-contacts. Recall that each contact gives a closed half-plane representing allowed positive rotations and another closed half-plane representing negative rotations. To constrain all motion and obtain form-closure, the set of allowable CORs must be empty; in other words, the intersection of the three closed half-planes for positive rotations must be empty and likewise the three closed half-planes for negative rotations. The following lemma gives a sufficient condition for the above which we exploit in our algorithms later. A set of vectors v_1, v_2, \dots, v_k is said to *positively span* a space Ξ if any vector $\xi \in \Xi$ can be written as $\sum_{i=1}^k \alpha_i v_i$ where the α_i are non-negative scalars. When $k = 3$ and $\Xi = \mathbf{R}^2$, it is easy to show that v_1, v_2, v_3

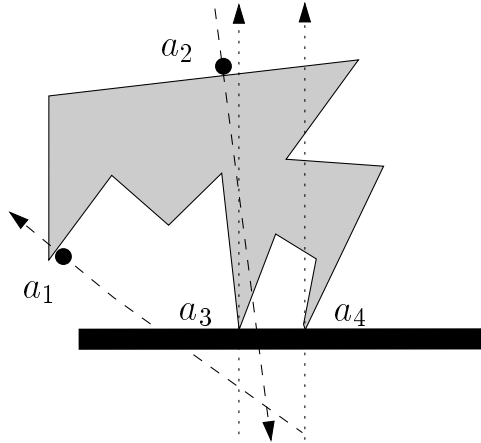


Figure 3: A polygon that is in form closure with one edge-contact and two point-contacts. The directed lines drawn indicate the constraints on the motion of the object by the individual contacts.

positively span \mathbf{R}^2 if and only if the angle between any two vectors not including the third is greater than zero and less than π .

Lemma 1 *An object P is in form-closure with point-contacts a_1, a_2 and edge-contact (a_3, a_4) if and only if*

1. *the three vectors along $l(a_1), l(a_2), l(a_3)$ positively span \mathbf{R}^2 , and*
2. *the intersection point of $l(a_1)$ and $l(a_2)$ lies in the interior of slab(a_3, a_4).*

See Fig. 3 for an example of an object that is in form closure with one edge-contact and two point-contacts. Notice that both conditions from Lemma 1 are satisfied. To compute one form-closure configuration, we look at maximal inscribed circles. Any maximal inscribed circle of a polygon without pairs of parallel edges touches the polygon in at least three points. The following result follows from Markenscoff et al. [11].

Lemma 2 [11] *For any maximal inscribed circle C of a polygon P without pairs of parallel edges, and c is the center of C , the three vectors $\overrightarrow{ca_i}$, $a_i \in (C(P) \cap \partial P)$, positively span \mathbf{R}^2 .*

Theorem 3 *Let P be a convex polygon with no pairs of parallel edges. P can be held in form-closure with one edge-contact and two point-contacts.*

Proof: Let $C(P)$ be the maximal inscribed circle of P with center c . Since P does not have parallel edges, there are always at least three intersection points of P and $C(P)$; the circle is tangent to the polygon at these points. Consider three

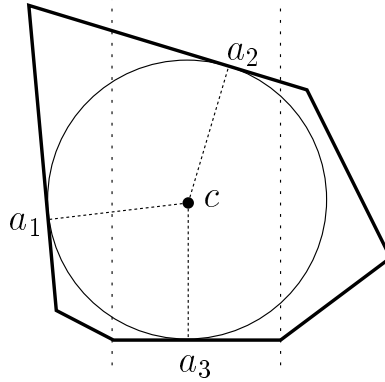


Figure 4: The tangent points of the largest inscribed circle of P with P give us the edge- and point-contacts for form-closure.

such points a_1, a_2 and b and their corresponding edges $E(a_1), E(a_2)$ and $E(b)$. Note that each of these points is strictly to the interior of the corresponding edge. Place one of these edges, say $E(b)$, against the edge-contact. The other two points, a_1, a_2 , are the point-contacts. Let the extreme points of intersection between the edge-contact and P be b_1, b_2 ; since P is convex, these are simply the ends of $E(b)$.

From Lemma 2 it follows that the vectors along bc, a_1c, a_2c positively span \mathbf{R}^2 . Since $l(a_1)$ is along a_1c , the same for a_2 , and the vector along bc is parallel to $l(b_1)$, the first condition for form-closure in Lemma 1 is satisfied.

As for the second condition, note that $l(a_1)$ and $l(a_2)$ are along the diameters of C and therefore intersect at the center c . Since $E(b)$ and C are tangent, the center c of C , lies strictly within the open region $slab(b_1, b_2)$. \square

Theorem 4 *Let P be an arbitrary polygon. If no edge of P is parallel to a different edge of its convex hull then P can be held in form-closure with one edge- and two point-contacts.*

Proof: The proof of this theorem is similar to that given for the previous theorem. Let $C(P)$ be a maximal inscribed circle of P and let c be its center. Consider three intersection points of $C(P)$ with ∂P : a_1, a_2 and a_3 . If none of these points is on one of the edges of the convex hull, grow $C(P)$ until it touches a edge e of the convex hull of P . Let this edge be bounded by points b_1, b_2 . Fix the edge-contact against e . Since none of the edges $E(a_1), E(a_2)$ or $E(a_3)$ is parallel to e , the angles between the lines $l(a_1), l(a_2)$ and $l(a_3)$ are all smaller than π . (In case some a_i is a concave vertex, $l(a_i)$ can be considered as any from a range of lines.) Thus, at least one of these lines (assumed directed from a_i into the object) crosses $slab(b_1, b_2)$ from left to right, say $l(a_1)$, and another crosses it from right to left, say $l(a_2)$. Place the point-contacts at a_1, a_2 . It is clear from the

definitions of $l(a_1)$ and $l(a_2)$ that $l(a_1), l(a_2), l(b_1)$ positively span \mathbf{R}^2 . Also, as before, c , the intersection point of $l(a_1)$ and $l(a_2)$ lies strictly within $slab(b_1, b_2)$. \square

The above proofs immediately gives us an algorithm to compute one form-closure for a given polygon. First, we compute the maximal inscribed circle of the polygon. This can be done in time $O(n \log n)$ by computing the medial axis of P [10] and in time $O(n)$ when P is convex. Then, if necessary, we compute the edge of the convex hull that is closest to this inscribed circle; this gives us the edge-contact. The two point-contacts can be determined as detailed above.

4 Generating Modular Fixtures

In this section we consider fixturing of the object with modular fixtures. We are given a rectangular rigid and flat surface into which circular holes of some precise dimension have been drilled to form a regular square lattice. Grid points are hole centers while *grid lines* are the vertical and horizontal lines through the grid points. Unit distance is defined as the distance between two grid points. Let P denote the given polygonal object with perimeter p and diameter d . The edge-fixel is assumed to be attached to the lattice parallel to one of the grid lines (it can be prefabricated along with the lattice). The horizontal or x -axis of our reference frame lies along the edge-fixel with P in the positive y half-plane. In addition to the edge-fixel, we are given a locator and a clamp whose placements are constrained by the grid. The precise constraints will be detailed later.

We present algorithms that output all valid fixtures. For ease of exposition and to give the intuition to the general case, we first make the assumption that the locator is a point and so is the entire clamp. The constraints imposed by the grid are that the point locator should be placed at a grid-point and the point clamp can exist anywhere along a grid-line. A simple procedure to enumerate all fixtures (non-output sensitively) is presented in Section 4.1. An improved algorithm is presented in Section 4.2 in which we formulate the problem in terms of data structure queries and give an efficient realization of the data structure ensuring output sensitivity. Finally, we give modifications required to relax the simplifying assumptions made and consider circular locators and rectangular clamp bodies and plungers in Section 4.3.

4.1 Enumerating all fixtures

Consider the polygon P shown bold in Fig. 5 resting with some particular edge of its convex hull against the edge-fixel, the thick bar shown at the bottom, attached to the lattice. As P may not be touching a grid-point on its boundary, we slide it, say to the left, until it does. Now we have a possible position for the locator. To compute all possible locator positions, it is sufficient to slide the polygon one unit

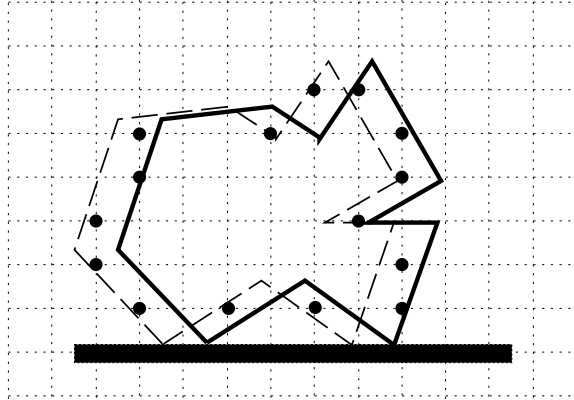


Figure 5: Sliding the polygon one unit along the edge-fixel to enumerate all feasible locator positions.

to the left and mark all grid-points encountered *during* the slide. Let $P = P(0)$ indicate the initial position of the polygon and $P(t)$, the polygon when shifted by t units to the left, $0 \leq t \leq 1$. Let t be termed the *shift variable*. A locator position L exists only at certain discrete values of the shift variable; at each one it contacts a different edge e of P . Let us term each such contact as a *locator-edge combination*. Let $t(L, e)$ denote the shift value when L touches e and $\alpha(L, e)$ the point of contact.

Clamp positions are those points on the boundary of P intersected by the grid lines in some $P(t)$. We distinguish between two cases: those points intersected by the horizontal grid lines are called *horizontal clamps* while the others are *vertical clamps*. Notice that since the shift is horizontal, horizontal clamp positions remain constant for all shift values and can be computed from the unshifted polygon. On the other hand, vertical clamp positions vary with the shift value. A vertical grid line V might intersect several edges e of P during the shift resulting in (*vertical*) *clamp-edge* combinations. Let the interval $(t_s(V, e), t_f(V, e))$ denote the range of shift values for which the clamp-edge combination of (V, e) exists; for t from within this interval, let $\alpha(V, e, t)$, denote the point of intersection between edge e in $P(t)$ and V .

There are at most $O(p)$ grid points (locator positions) generated by this process (a better bound is $O(\min(d^2, p))$, where d is the diameter of P , but we'll assume $p < d^2$, without loss of generality). Further, the total number of locator-edge and clamp-edge combinations is $O(p + n)$ and can be computed in $O(n + p)$ time.

A naive procedure to generate all valid fixtures is therefore to consider each locator-edge combination and clamp-edge combination pair, check if they simultaneously exist (w.r.t. the shift variable; this is necessary only for locators with vertical clamps) and if so, test if they achieve form-closure. Iterating over all

convex hull edges against the edge-fixel, this gives a $\Theta(n(p+n)^2)$ algorithm. In Section 4.2 we improve this to $O(n(p+n)^{4/3+\epsilon} + K)$ using efficient data structures.

Remark 1 In order for all the possible fixtures to be feasible, the lattice should have dimensions at least $\lceil d+1 \rceil \times \lceil d+1 \rceil$. A lattice of these dimensions also suffices. In the general case of locators and clamps of some non-zero diameters d_l, d_c , the sum $d + d_c + d_l$ should replace d .

Remark 2 The user may specify that certain locations on the boundary of P or that certain access-directions to the object be fixel-free. Edge-fixel, locator and clamp locations can be pruned out by checking each for violation. This will not affect the total time bound.

4.2 Efficient data structures

In the following we'll generate fixture configurations in which the locator is at the left side of the object. The procedure should be repeated for the other symmetrical case. The case of horizontal clamps is easier and gives some intuition to the vertical case and therefore is treated first.

Fixtures with a locator and horizontal clamp As mentioned above, horizontal clamp positions can be determined by intersecting the horizontal grid lines with the object and they remain constant for all shifts. Thus the half-plane constraint corresponding to each horizontal clamp can be represented by a directed line at the point of contact and normal to the edge of contact, directed into the object (see Section 2). (If the point of contact is a concave vertex, we choose a suitable directed line that points "into" a cone of possible directions as discussed before. Convex vertices are not feasible clamp or locator positions.) Since we are interested only in clamps to the right of the object (because locators are to the left), only directed lines crossing the slab from right to left need to be included. There are $O(n+p)$ such directed lines. Call these the *clamp lines*.

Another set of directed lines comes from the locators. For each locator-edge combination (L, e) , create a similar directed line at the contact point $\alpha(L, e)$. It is not necessary to take into consideration the shift value $t(L, e)$ at which the combination (L, e) exists because every horizontal clamp always exists and can exist simultaneously with any single locator irrespective of $t(L, e)$. Again, there are $O(n+p)$ *locator lines*. It is sufficient to consider directed lines crossing the slab from left to right.

Thus we obtain two sets of directed lines defining half-plane constraints on P ; one set of lines associated with possible clamp positions and one set of lines for the possible locator positions. We want to find all combinations of lines, one from each set, such that Conditions 1 and 2 in Lemma 1 are satisfied. Intersecting the

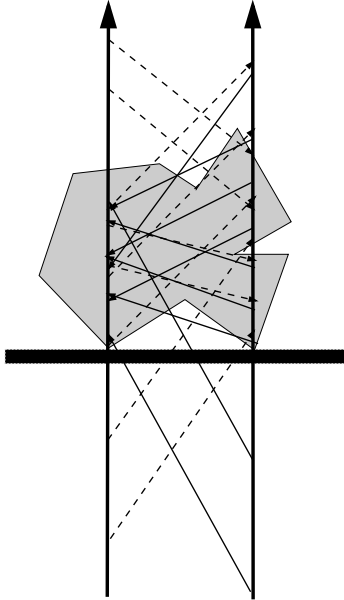


Figure 6: The directed line segments (directed lines representing half-plane constraints restricted to the slab formed by the edge-fixel at the bottom) resulting from the object in Figure 5. The dotted line segments (from left to right) correspond to the locators while the solid lines (from right to left) are from the clamps.

directed (infinite) lines with the slab corresponding to the fixed convex hull edge of P , we obtain two sets of directed line *segments*.

Given a locator line segment, we wish to detect, in an output-sensitive manner, all clamp line segments that (properly) intersect it and such that the vectors along the two segments and the upward vertical vector positively span \mathbf{R}^2 . See Fig. 6. Formally, the basic query that we want to answer is:

QUERY A Let S be a set of m directed line segments, each with an end-point along two vertical lines (the “slab”), and each with orientation from the open interval $(-\frac{1}{2}\pi, \frac{1}{2}\pi)$. Store S such that for a query segment q also similarly anchored on the slab but with orientation in the interval $(\frac{1}{2}\pi, \frac{3}{2}\pi)$ one can report all segments $s \in S$ such that

1. the vectors along s, q and the upward vertical vector positively span \mathbf{R}^2 , and
2. s and q intersect strictly inside the slab.

Lemma 5 *There exists a data structure that solves QUERY A in time $O(\log^2 m + k)$ time, where k is the number of reported segments. Preprocessing takes $O(m \log^2 m)$*

time.

Proof: Since the orientation of all segments in S is in the interval $(-\frac{1}{2}\pi, \frac{1}{2}\pi)$, and the query segment q is oriented in direction $\alpha \in (\frac{1}{2}\pi, \frac{3}{2}\pi)$, it follows that all the line segments oriented in the subinterval $(\frac{3}{2}\pi, \alpha + \pi)$ satisfy the first condition in the query (this is because the angles between any two of the vectors not including the third are all smaller than π). The data structure that we need to compute this set of segments is a range tree on the angles of S ; a query on this tree in order to find all segments in the desired interval takes $(\log m + k')$ time [14]. The k' angles can be reported in $O(\log m)$ canonical subsets in the nodes of the tree. In these subsets we have to search for intersections between the corresponding line segments and the query line segment. We do this by associating an intersection query structure with every node in the segment tree. This secondary structure is built on all the line segments that are associated with the angles in this node. Such an intersection query in a slab reduces to a dominance range query problem as will be described in the full paper. A query on this range query structure takes $O(\log m + k'')$ time, k'' being the number of answers. Combining these two structures gives us a query time of $O(\log^2 m + k)$, where k is the number of reported segments. The structure can easily be reported in time $O(m \log^2 m)$. \square

Now we are ready to prove the following theorem.

Theorem 6 *Let P be a polygon with perimeter p and n edges. All configurations holding P in form-closure using one edge-fixel, one locator and one horizontal clamp can be enumerated in time $O(n(n+p)\log^2(n+p) + K)$, where K is the number of form-closure configurations.*

Proof: First compute the convex hull of P . For every edge of the convex hull along the edge-fixel do the following. Compute all the $O(n+p)$ locator lines and all the $O(n+p)$ clamp lines. This can be done in $O(n+p)$ time. For every locator line, we perform *QUERY A*. In all, this takes takes $O((n+p)\log^2(n+p) + k')$ time, since there are at most $O(n+p)$ of these lines; k' is the number of fixture configurations in this iteration of the algorithm. The total time therefore becomes $O(n(n+p)\log^2(n+p) + K)$. \square

In contrast to querying the locator lines one at a time, we can “batch” together all the locator and horizontal clamp lines and by processing them together carefully, it is possible to improve the above time complexity to $O(n(n+p)\log(n+p) + K)$. However, we chose to retain the query algorithm since it gives some intuition towards handling vertical clamps.

Fixtures with a locator and vertical clamp As before, assume that one edge of the convex hull of P is flush against the edge-fixel. Shifting the polygon

one unit to the right generates all locator-edge combinations along with a particular shift value at which each exists. However, in contrast to horizontal clamps, a vertical grid line V does not always touch an edge e of $P(t)$ at the same spot during the shift, *i.e.*, the point of contact $\alpha(V, e, t)$ is a linear function of t . Also vertical clamp-edge combinations only exist for a subinterval of shift values $(t_s(V, e), t_f(V, e))$. For these reasons, the algorithm to compute form-closures using vertical clamps is different from the algorithm for horizontal clamps described above.

The major modification required is to explicitly handle shift by a third dimension. Firstly, the slab corresponding to the edge-fixel is extended into the t dimension by one unit to form a rectangular block. The directed line segment representing the half-plane constraint from a locator-edge combination (L, e) is lifted to the $t = t(L, e)$ plane. For vertical clamp-edge combinations (V, e) , we need to represent the half-plane constraint at *every* shift instant in $(t_s(V, e), t_f(V, e))$. Fortunately, since e is linear, the corresponding directed line-segments are parallel to each other (perpendicular to e); the directed line at instant t has to be perpendicular to edge e in $P(t)$ and passing through $\alpha(V, e, t)$. This continuum of parallel directed line segments thus forms a parallel band between $t_s(V, e)$ and $t_f(V, e)$; the band is oriented in the direction of the lines comprising it.

In this representation, we look for proper intersections of oriented bands corresponding to vertical clamp-edge combinations with directed lines corresponding to locator-edge combinations that further satisfy the condition related to positively spanning \mathbf{R}^2 . By restricting attention to the rectangular block, we get a parallelogram for every band and a directed line segment for every directed line.

As before we can consider right-to-left parallelograms to be intersected by left-to-right directed line-segments. The other symmetrical case can be treated in the same fashion.

QUERY B

Let S be a set of m directed line segments anchored between two parallel faces of a rectangular block as described above and each parallel to the xy plane. Store S such that for an anchored oriented parallelogram q between heights t_a and t_b as detailed above one can report all segments $s \in S$ such that

1. the vectors along s and q and the vertical vector in positive y -direction positively span \mathbf{R}^2 and
2. s and q intersect strictly inside the rectangular block.

Lemma 7 *There exists a data structure that solves QUERY B in time $O(m^{1/3+\epsilon} + k)$ time for any positive constant ϵ , where k is the number of reported segments. The preprocessing time is $O(m^{4/3+\epsilon})$.*

Proof: The data structure consists of a three-level tree. The first level is used to detect the segments of S that have the correct angle. The second level detects all

segments that can possibly intersect the query parallelogram, because they exist at a time $t \in (t_a, t_b)$. The third layer is used to check for intersections between a query plane and a set of segments. First, we build a range tree on the orientations of the segments in S , which is the same as the one that we used for *QUERY A*, in order to satisfy Condition 1. Since the parallelogram only exists between shifts (t_a, t_b) which may not be $(0, 1)$. We build a range tree representing heights at which the segments occur. With every node in this tree, we associate a partition tree [1, 7] on the segments associated with the values that are present below this particular node. Such a partition tree can be queried with a plane in order to obtain all segments that intersect this query plane. Now we query the range tree with (t_a, t_b) in order to obtain all segments in between those shifts. Having done this, we can replace the parallelogram by the infinite plane containing it (because we only consider the segments extracted within the shifts of the parallelogram). In the corresponding partition trees associated with the $O(\log m)$ canonical subsets in the nodes bordering the search path in the range tree, we search for the segments that intersect the query plane. This takes $O(m^{1/3+\epsilon} + k)$ per query, where k is the number of answers. The logarithmic time that we need for the two queries on the range trees are subsumed by the m^ϵ factor. \square

Theorem 8 *Let P be a polygon with perimeter p and n edges. All configurations holding P in form-closure using one edge-fixel, one locator and one vertical clamp can be enumerated in time $O(n(p+n)^{4/3+\epsilon} + K)$.*

Proof: We iterate over every edge of the convex hull of P against the edge-fixel. For a particular convex hull edge, the time to compute the vertical clamp-edge and locator-edge combinations as parallelograms and directed line-segments, respectively is $O(p+n)$. We query every parallelogram corresponding to a vertical clamp-edge combination by performing *QUERY B*; this takes $O((p+n)^{4/3+\epsilon} + k')$ time in all, since there are at most $(p+n)$ parallelograms; k' is the number of solutions for the current iteration. The total time complexity of the algorithm is $O(n(p+n)^{4/3+\epsilon} + K)$, since we have to perform this for every edge of the convex hull; K is the total number of form-closure configurations. \square

4.3 Large locators and clamps

In the previous subsection, we assumed that the locator and clamp fixels were points and that clamps could exist anywhere along a grid line. In this section we consider the more realistic case of large locators and clamps. To consider a specific model, we assume the following. Locators are circular of some constant radius r and the clamp body is a rectangle. Perpendicular to one of its sides emanates the plunger which is also rectangular with a semicircular tip. The locations of the clamp body and the locator are constrained by grid points in

that a reference point on each has to coincide with a grid point. Finally, the clamp can be configured in an axis-parallel fashion to allow either horizontal (horizontal clamp) or vertical (vertical clamp) motion of the plunger. We assume the maximal plunger extension is one grid unit.

To handle locators, we simply compute the Minkowski difference of the input polygon P with a circle of radius r to get a grown object P_r bounded by straight lines and circular arcs of radius r at convex vertices. This effectively shrinks the locators to points. Now we do as before: shift P_r by one unit to the left and identify the set of grid points crossed over during the shift. We can ignore the circular arcs in P_r as we perform the shift (this is because we cannot place locators at convex vertices). Although each grid point can be crossed by several edges, there are only $O(n + p)$ locator-edge combinations in all, each with an associate shift value. Further, these are computable in $O(p + n \log n)$ time (given the polygon edges in order). Now we go back to the original polygon P and radius r locators. For each locator-edge combination, we compute the directed line representing the half-plane constraint. In all, we obtain a set S_L of $O(n + p)$ such directed lines, each associated with a shift value.

For horizontal clamps, we first compute the Minkowski difference between the polygon and the horizontally configured clamp body. The resulting object is shifted by one unit as before and all grid-points that are at most one unit horizontally from the object are collected. These are locations where (the reference point on the) clamp body can be positioned to avoid collisions with P for certain intervals of the shift variable. However, unlike the case of point clamps treated before in which a clamp-edge combination was valid for all shifts, in the present case, because of the clamp body constraint, there can be several (disjoint) subintervals of valid shifts for a particular clamp-edge combination (See Fig. 7). By careful counting, it can be shown that the number of subintervals over all clamp-edge combinations totals $O(n + p)$. Each horizontal clamp-edge combination results in a directed line representing the corresponding half-plane constraint; this directed line is labelled with the appropriate subintervals for which the clamp-edge combination exists. To simplify matters, let us suppose that if there are β disjoint subintervals of $[0, 1]$ during which a clamp-edge combination exists, then we have β copies of the appropriate directed line, each copy representing a different disjoint subinterval. Thus, in all, we have a set S_H of $O(n + p)$ directed lines, each associated with a *single* interval of shift values.

Theorem 9 *Let P be a polygon with perimeter p and n edges. All configurations holding P in form closure using an edge-fixel, a round locator and a horizontal clamp can be enumerated in time $O(n(n + p) \log^3(n + p) + K)$, where K is the number of configurations.*

Proof: From the above discussion, the polygon P results in a set S_L of $O(n + p)$ directed lines each associated with a single shift value and another set S_H of $O(n + p)$ directed lines associated with single intervals of shift values. These sets

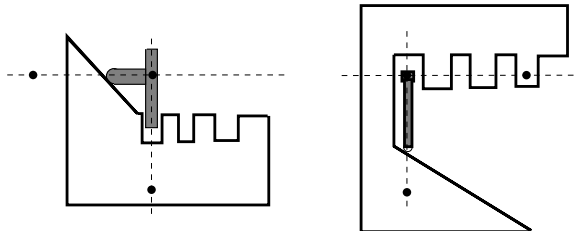


Figure 7: Several subintervals of valid shifts for a single clamp-edge combination. The figure on the left shows the case of a horizontal clamp while that on the right is a vertical clamp. Clamps are shown shaded.

can be computed in time $O(p+n \log n)$. We can build a range tree on the shift values in S_L and query it with each subinterval (a, b) from S_H to extract all directed lines from S_L whose shift value lies in (a, b) . With this tree we can associate a secondary structure as described in the algorithm in the previous subsection. This increases the query time for one element of S_H to $O(\log^3(n+p) + k)$, where k is the number of valid fixtures with this edge-clamp position combination. The total time complexity of the algorithm is therefore as claimed; K is the total number of valid fixtures. The preprocessing time is dominated by the total time of the algorithm. \square

For vertical clamps, much of the initial analysis is the same. We compute the Minkowski difference between the polygon and the vertically configured clamp body. The resulting object is shifted by one unit and all grid-points that are at most one unit vertically from the object are locations where (the reference point of the) clamp body can be positioned freely without collisions for certain intervals of the shift variable. Fig. 7 shows that a single clamp-edge combination can exist over several (disjoint) subintervals. Also, the total number of such subintervals, taken over all clamp-edge combinations, is at most $O(n+p)$. However, unlike the case of horizontal clamps, a vertical clamp tip contacts different points of the edge for different shifts. Thus a single directed line does not suffice but a continuum of directed lines, or a parallelogram in the (x, y, t) space is required. Every subinterval of every clamp-edge combination results in a different parallelogram; thus there is a set S_V of $O(n+p)$ parallelograms, that can be computed in time $O(p+n \log n)$. From the above discussion and the proof of Theorem 8 we can conclude:

Theorem 10 *Let P be a polygon with perimeter p and n edges. All configurations holding P in form closure using an edge-fixel, a round locator and a vertical clamp can be enumerated in time $O(n(n+p)^{4/3+\epsilon} + K)$, where K is the number of configurations.*

Intersections between clamp and locator We have so far taken into account collisions between locators and the polygon as also clamps with the polygon. It could also happen that the locator and clamp intersect in certain configurations. However, recall that the clamp and the locator have a constant size. We can therefore check intersections between these two fixels in a generated solution in constant time, and therefore in $O(K)$ time over all the solutions. The number of wrong solutions can be bounded by $O(n + p)$, hence this does not change the time complexity of the algorithms.

Clamp plunger extension We have assumed so far that the maximum plunger extension $s = 1$. For general s , the clamp body positions are those within s units of a shifted polygon. The number of possible locations increases by a multiplicative factor of s . In other words, replace $n + p$ by $(n + p)s$ in the time complexity for the algorithm. Increased clamp extension capability can be useful if clamp bodies are large and have to be placed outside the convex hull of the polygon; a long plunger then might be able to reach into a deep cavity while a short one couldn't.

5 Discussion

In this paper we considered a new class of contacts called edge-contacts and showed that almost all polygonal objects can be fixtured by an edge- and two point-contacts. Based on the edge-contact, we proposed an edge-fixel for use in modular fixture design. In the full version of the paper, we will prove that all rectilinear polygons with edge length at least $2\sqrt{2}$ grid units that are not rectangular in convex hull can be modular fixtured in our model. We presented an output-sensitive algorithm that computes all valid edge-locator-clamp fixtures on polygonal objects. By feeding these solutions through a given quality metric or Boolean comparator, an *optimal* fixture may be computed.

We are currently investigating fixturing an object with two perpendicular edge-fixels and one point-fixel. We also wish to look ahead to non-convex curved objects. A promising direction of research appears to be considering T-slot clamps [17]. Here, clamping is done by a fixel traveling in a “T”-shaped slot. Such fixels can translate in both vertical and horizontal direction along the grid lines. We believe that similar techniques can be applied here.

References

- [1] Pankaj K. Agarwal and J. Matoušek. Ray shooting and parametric search. *SIAM J. Comput.*, 22(4):794–806, 1993.

- [2] H. Asada and A. B. By. Kinematic analysis of workpart fixturing for flexible assembly with automatically reconfigurable fixtures. *IEEE Journal of Robotics and Automation*, RA-1(2), June 1985.
- [3] J. J. Bausch and K. Youcef-Toumi. Kinematic methods for automated fixture re-configuration planning. In *International Conference on Robotics and Automation*. IEEE, May 1990.
- [4] W. Boyes, editor. *Handbook of Jig and Fixture Design, 2nd Edition*. Society of Manufacturing Engineers, 1989.
- [5] R. C. Brost and K. Y. Goldberg. A complete algorithm for synthesizing modular fixtures for polygonal parts. In *International Conference on Robotics and Automation*. IEEE, May 1994.
- [6] Y-C. Chou, V. Chandru, and M. M. Barash. A mathematical approach to automatic configuration of machining fixtures: Analysis and synthesis. *Journal of Engineering for Industry*, 111, November 1989.
- [7] M. de Berg. *Ray Shooting, Depth Orders and Hidden Surface Removal*, volume 703 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, Germany, 1993.
- [8] M. V. Gandhi and B. S. Thompson. Automated design of modular fixtures for flexible manufacturing systems. *Journal of Manufacturing Systems*, 5(4), 1986.
- [9] E. G. Hoffman. *Modular Fixturing*. Manufacturing Technology Press, Lake Geneva, Wisconsin, 1987.
- [10] D. T. Lee. Medial axis transformation of a planar shape. *IEEE Trans. Pattern Anal. Mach. Intell.*, PAMI-4:363–369, 1982.
- [11] X. Markenscoff, L. Ni, and C.H. Papadimitriou. The geometry of grasping. *IJRR*, 9(1), February 1990.
- [12] B. Mishra. Workholding: Analysis and planning. In *International Conference on Intelligent Robots and Systems*. IEEE/RSJ, July 1991.
- [13] B. Mishra, J. T. Schwartz, and M. Sharir. On the existence and synthesis of multifinger positive grips. *Algorithmica*, 2(4):641–558, 1987.
- [14] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, New York, NY, 1985.
- [15] F. Reuleaux. *The Kinematics of Machinery*. Macmillan and Company, 1876. Republished by Dover in 1963.
- [16] A. Wallack and J. Canny. Planning for modular and hybrid fixtures. In *International Conference on Robotics and Automation*. IEEE, May 1994.
- [17] Y. Zhuang, K.Y. Goldberg, and Y-C. Wong. On the existence of modular fixtures. In *International Conference on Robotics and Automation*, pages 543–549. IEEE, May 1994.