Optimized Synthesis of Snapping Fixtures*

Tom Tsabar[1], Efi Fogel[1], and Dan Halperin[1]

The Blavatnik school of Computer Science, Tel Aviv University, Israel
{tomtsabar9@gmail.com,efifogel@gmail.com,danha@post.tau.ac.il}

Abstract. Fixtures for constraining the movement of parts have been extensively investigated in robotics, since they are essential for using robots in automated manufacturing. This paper deals with the design and optimized synthesis of a special type of fixtures, which we call *snapping fixtures*. Given a polyhedral workpiece P with n vertices and of constant genus, which we need to hold, a snapping fixture is a semi-rigid polyhedron G, made of a palm and several fingers, such that when P and G are well separated, we can push P toward G, slightly bending the fingers of G on the way (exploiting its mild flexibility), and obtain a configuration, where G is back in its original shape and P and G are inseparable as rigid bodies. We prove the minimal closure conditions under which such fixtures can hold parts, using Helly's theorem. We then introduce an algorithm running in $O(n^3)$ time that produces a snapping fixture, minimizing the number of fingers and optimizing additional objectives, if a snapping fixture exists. We also provide an efficient and robust implementation of a simpler version of the algorithm, which produces the fixture model to be 3D printed and runs in $O(n^4)$ time. We describe two applications with different optimization criteria: Fixtures to hold add-ons for drones, where we aim to make the fixture as lightweight as possible, and small-scale fixtures to hold precious stones in jewelry, where we aim to maximize the exposure of the stones, namely minimize the obscuring of the workpiece by the fixture.

Keywords: Computational Geometry, Automation, Grasping, Fixture Design

1 Introduction

A fixture is a device that holds a part in place. Constraining the movement of parts is a fundamental requirement for using robots in automated manufacturing [1],[2, Section 3.5]. There are many types and forms of fixtures; they range from modular fixtures synthesized on a lattice to fixtures generated to suit a specific part. A fixture possesses some grasp characteristics. For example, a grasp with complete restraint prevents loss of contact, prevents any motion, and thus may by considered secure. Two primary kinematic restraint properties are *form*

* This work has been supported in part by the Israel Science Foundation (grant nos. 825/15,1736/19), by the Blavatnik Computer Science Research Fund, and by grants from Yandex and from Facebook.

closure and *force closure* [3, Chapter 38]. Both properties guarantee maintenance of contact under some conditions. However, the latter typically relies on contact friction; therefore, achieving force closure typically requires fewer contacts than achieving form closure. Fixtures with complete restraint are mainly used in manufacturing processes where preventing any motion is critical. Other types of fixtures can be found anywhere, for example, in the kitchen where a hook holds a cooking pan, or in the office where a pin and a bulletin board hold a paper still. This paper deals with a specific problem in this area; here, we are given a rigid object, referred to as the *workpiece*, and we seek for an automated process that designs a semi-rigid object, referred to as the snapping *fixture*, such that, starting at a configuration where the workpiece and the holding fixture are separated, they can be pushed towards each other, applying a linear force and exploiting the mild flexibility of the fixture, into a configuration where both the workpiece and the fixture are inseparable as rigid bodies. A generated fixture has a base part, referred to as the *palm*, and fingers connected to the palm; see Section 2.1 for formal definitions. Without additional computational effort, a hook, a nut, or a bolt can be added to the palm resulting in a generic fixture that can be utilized in a larger system. Another advantage of the single-component flexible fixture is that it can easily be 3D-printed. We have 3D-printed several fixtures that our generator has automatically synthesized for some given workpieces. The objective of the algorithm is obtaining snapping fixtures with the minimal number of fingers. With additional care that also accounts for properties of the material used to produce the fixtures, the smallest or lightest possible fixture can be synthesized, for a given workpiece. This can (i) expedite the production of the fixture using, e.g., additive manufacturing, (ii) minimize the weight of the produced fixture, and (iii) maximize the exposed area of the boundary of the workpiece when held by the fixture.

1.1 Background

Form closure has been studied since the 19th century. Early results showed that at least four frictionless contacts are necessary for grasping an object in the plane, and seven in 3D space. Specifically, it has been shown that four and seven contacts are necessary and sufficient for the form-closure grasp of any polyhedron in the 2D and 3D case, respectively [4, 5].

Automatic generation of various types of fixtures, and in particular, the synthesis of form-closure grasps, are the subjects of a diverse body of research. Brost and Goldberg [6] proposed a complete algorithm for synthesizing modular fixtures of polygonal workpieces by locating three pegs (locators), and one clamp on a lattice. Their algorithm is complete in the sense that it examines all possible fixtures for an input polygon. Their results were obtained by generating all configurations of three locators coincident to three edges, for each triplet of edges in the input polygon. For each such configuration, the algorithm checks whether *form closure* can be obtained by adding a single clamp. Our work uses a similar strategy to obtain all possible configurations. In subsequent work Zhuang, Goldberg, and Wong [7] showed that there exists a non-trivial class of polygonal workpieces that cannot be held in form closure by any fixture of this type (namely, a fixture that uses three locators and a clamp). They also considered fixtures that use four clamps, and introduced two classes of polygonal workpieces that are guaranteed to be held in form closure by some fixture of this type. Wallack and Canny [8] proposed another type of fixture called the vise fixture and an algorithm for automatically designing such fixtures. The vise fixture includes two lattice plates mounted on the jaws of a vise and pegs mounted on the plates. Then, the workpiece is placed on the plates, and *form closure* is achieved by activating the vise and closing the pins from both sides on the workpiece. The main advantage in this type of fixture is its simplicity of usage. Brost and Peters [9] extended the approach exploited in [6] to three dimensions. They provided an algorithm that generates suitable fixtures for three-dimensional workpieces. Wagner, Zhuang, and Goldberg [10] proposed a three-dimensional seven-contact fixture device and an algorithm for planning form-closure fixtures of a polyhedral workpiece with pre-specified pose. A summary of the studies in the field of flexible fixture design and automation conducted in the last century can be found in [11]. Related studies in the field of grasping and manipulation are summarized in [12]. Subsequent works studied other types of fixtures and provided algorithms for computing them, for example, unilateral fixtures [13], which are used to fix sheet-metal workpieces with holes. Other studies focused on grasping synthesis algorithms with autonomous robotic *fingers*, where a single robotic hand gets manipulated by motors and being used to grasp different workpieces; an overview of such algorithms can be found in [14]. A common dilemma for all the grasping and fixture design algorithms is defining and finding the optimal grasp. Several works, e.g., [15] and [16], discuss such quality functions and their optimization.

1.2 Our Results

We introduce certain properties of minimal snapping fixtures of given workpieces. Formally, we are given a closed polyhedron P of complexity n and of a constant genus that represents a workpiece. The surface of a polyhedron of genus zero is homeomorphic to a sphere. In our work we allow more complicated polyhedra; see, for example, Figure 4a.[1] In our analysis in the sequel we assume that the genus is bounded by a constant. We introduce an algorithm that determines whether a closed polyhedron G that represents a fixture exists, and if so, it constructs it in $O(n^3)$ time. This significantly improves our simpler $O(n^4)$ algorithm [17]. We also provide an efficient and robust implementation of the latter. In addition, we present two practical cases that utilize our implemented algorithm: One is the generation of a snapping fixture that mounts a device to an unmanned aerial vehicle (UAV), such as a drone. The other is the generation of a snapping fixture that mounts a precious stone to a jewel, such as a ring.

[1] The genus counts the number of "handles" in the polyhedron; see, e.g., https://mathworld.wolfram.com/Genus.html.

The common objective in both cases is, naturally, the firm holding of the workpiece. In the first case, we are interested in a fixture with minimal weight. In the second case we are interested in a fixture that minimally obscures the precious stone. We are not aware of similar work on semi-rigid one-part fixtures; thus, we do not conduct any comparisons, but we provide some benchmark numbers we have obtained while executing our generator. Note that, in theory, the generated fixtures prevent any linear motion, but do not necessarily prevent angular motion; however, fixtures that do not posses the *form closure* property are rarely obtained in practice. Handling angular motion is left for future research [17].

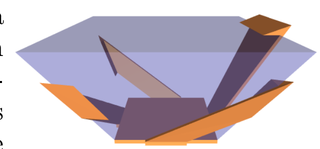
1.3 Outline

The rest of this paper is organized as follows. Terms and definitions for our snapping fixtures and theoretical bounds and properties are provided in Section 2. The synthesis algorithm is described in Section 3 along with the analysis of its complexity. Two applications are presented in Section 4. Finally, we report on experimental results in Section 5. Our full paper [17] contains a notation glossary, proofs of several lemmas, corollaries, observations, and a theorem, some limitations of our generator, and suggestions for future research.

2 Terminology and Properties

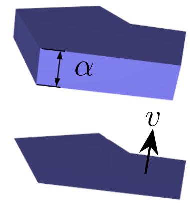
In this section we describe the structure and properties of our snapping fixtures.

2.1 Fixture Structure



Consider an input polyhedron P that represents a workpiece, such as the one transparently rendered in blue in the figure to the right. The structure of a fixture of P, rendered in orange in the figure, resembles the structure of a hand; it is the union of a single polyhedral part referred to as the *palm*, several polyhedral parts, referred to as *fingers*, which are extensions of the *palm*, and semi-rigid joints that connect the palm and the fingers. Each *finger* consists of two polyhedral parts, namely, *body* and *fingertip*, and the semi-rigid joint between the *body* and the *fingertip*. The various parts, i.e., palm, bodies, and fingertips, are disjoint in their interiors. In the following we describe these parts in detail.

Definition 1 (α-extrusion of a polygon and base polygon of an α-extrusion). *Let L denote a polygon in space, let v denote a normal to the plane containing L, and let v_α denote the normal scaled to length α. The α-*extrusion *of L is a polyhedron Q in space, which is the extrusion of L along v_α. The polygon L is referred to as the* base polygon *of Q; see the figure below.*



We use the abbreviation α-extrusion of a facet f of some polyhedron to refer to the α-extrusion Q of the geometric embedding of the facet f, and we refer to the facet of Q that overlaps with f as the base facet of the α-extrusion Q.

Our formal computational model is oblivious to the thickness of the various parts. In this model the parts are flat and if two parts are connected by a joint, they share an edge, which is the axis of the joint. Our generator, though, synthesizes solid models of fixtures. We use α-extrusion to inflate the various parts.

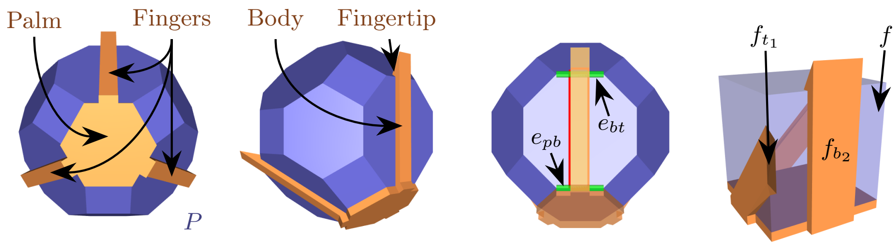
Let G denote a snapping fixture made of a palm, k fingers $F_1, F_2, \ldots, F_k$, and corresponding joints. The palm is an α_p-extrusion of a facet f_p of the workpiece P. (The various α values are discussed below.) Consider a specific finger $F = F_i$ of G. The body of F is defined by one of the neighboring facets of f_p, denoted f_b. The fingertip of F is defined by one of the neighboring facets of f_b, denoted f_t, $f_t \neq f_p$. Let e_{pb} denote the common edge of f_p and f_b, and let e_{bt} denote the common edge of f_b and f_t. Note that in some degenerate cases e_{pb} and e_{bt} are incident to a common vertex. The body of a finger is an α_b-extrusion of f_b. Let v denote the cross product of the vector that corresponds to e_{bt} and the normal to the plane containing f_t of length α_t. Let q_t denote the quadrilateral defined by the two vertices incident to e_{bt} and their translations by v. The fingertip is an α_t-extrusion of q_t. The axis of the joint that connects the palm and the body of F coincides with e_{pb} and the axis of the joint that connects the body of F with its fingertip coincides with e_{bt}. The value α_p and the values α_b and α_t for each finger determine the trade-off between the strength and flexibility of the joints.[2] They depend on the material and shape of the fixture. In our implementation they can be determined by the user.[3]

For a complete view of a workpiece and a snapping fixture consider Figure 1. Observe that both the palm and the fingers of the fixture in the figure differ from the formal definitions above. The differences stem from practical considerations. In particular, the parts in the figure have smaller volumes, which (i) reduces fabrication costs, and (ii) resolves collision between distinct fingers. In some degenerate cases (see Figure 1d) distinct fingers could have overlapped. In the figure, the base facet of the fingertip of one finger, f_{t_1}, coincides with f, a facet of the workpiece. Likewise, the base facet of the body of the other finger, f_{b_2}, also coincides with f. Avoiding overlaps is achieved by simultaneously shrinking the base facets f_{t_1} and f_{b_2}. Now, the fingertip grips only the tip of f and the body is stretching only on a small portion of the workpiece facet. As another example, consider the body of a finger depicted in Figure 1(c); it is the α_b-extrusion of a quadrilateral defined by two points that lie in the interior of e_{pb} and two points that lie in the interior of e_{bt}, as opposed to the formal definition above, where the body is the α_b-extrusion of the entire facet of P. Also, in reality, parts are not fabricated separately, and the entire fixture is made of the same flexible material. Instead of rotating about the joint axes, the entire fingers bend. The

[2] Typically, these values are identical.

[3] For example, in several of the fixtures that we produced, they were set to 5mm.

differences, though, have no effect on the correctness of the proofs and algorithm (which adhere to the formal definitions) presented in the sequel. These structural changes and the extrusion values, merely determine the degree of flexibility and strength of the fixture [17].



(a) (b) (c) (d)

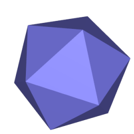
Fig. 1. (a), (b), (c) Different views of a truncated cuboctahedron (blue) and a snapping fixture (orange). (d) A transparent cube (blue) and a snapping fixture (orange).

2.2 The Configuration Space

The workpiece and its snapping fixture form an assembly. Each joint in the fixture connects two parts; it enables the rotation of one part with respect to the other about an axis. Each joint adds one degree of freedom (DOF) to the configuration space of the assembly.

In our context, the workpiece and its snapping fixture are considered assembled, if they are *infinitesimally inseparable*. When two polyhedra are infinitesimally inseparable, any linear motion applied to one of the polyhedra causes a collision between the polyhedra interiors. The workpiece and the fixture are in the *serving configuration* if (i) they are separated (that is, they are arbitrarily far away from each other), and (ii) there exists a vector v, such that when the fixture is translated by v, as a result of some force applied in the direction of v, exploiting the flexibility of the joints of the fixture, the workpiece and the fixture become assembled. When the workpiece and its snapping fixture are separated, the fixture can be transformed without colliding with the workpiece to reach the serving configuration.[4]

2.3 Spreading Degree



The *spreading degree* is the number of facets involved in the definition of a finger. In this paper we restrict ourselves to snapping fixtures that have fingers with spreading degree two, which means that the body of every finger is based on a single facet of P. Every finger (the body and the fingertip) stretches over two facets of P. Naturally, fingers with a higher spreading-degree reach further. An icosahedron, for example, (depicted in the figure above) does not admit a valid fixture with spreading degree two. This is proven by exhaustion running our implemented algorithm.

[4] The video clip available at http://acg.cs.tau.ac.il/projects/ossf/snapping_fixtures.mp4 illustrates the snapping operation.

2.4 Fixture Planning

The basic objective of our fixture algorithms is obtaining fixtures with the minimal number of fingers. Our generator is of the exhaustive type. As explained in Section 3, it examines many different possible candidates of fingers, before it reaches a conclusion. The simple (and implemented) algorithm, for example, visits every valid fixture (of 2, 3, or 4) fingers; thus, it can be used to produce all or some valid fixtures according to any combination of optimization criteria. As aforementioned, the generator synthesizes fixtures of spreading degree two. Extending the generator to enable the synthesis of fixtures with an increased spreading degree (without further modifications) will directly increase the search space exponentially.

2.5 Properties

Definition 2 (unit circle, semicircle, open semicircle). *An open semicircle is a semicircle excluding its two endpoints. An open hemisphere is a hemisphere excluding the great circle that comprises its boundary curve.*

Definition 3 (Covering set). *Let* $\mathcal{S} = \{s_1, ..., s_{|\mathcal{S}|}\}$ *be a finite set of subsets of* $\mathbb{R}^d$ *and* C *be a set of points in* $\mathbb{R}^d$. *If* $\bigcup_{i=1}^{|\mathcal{S}|} s_i \supseteq C$ *then* $\mathcal{S}$ *is a covering set of* C.

A pair of open unit semicircles (respectively, hemispheres) are called antipodal if the closure of their union is the entire unit circle (respectively, sphere).

For lack of space, we defer portions of the formal analysis to the full paper [17]. In particular a sequence of lemmas proved in [17] yield the following corollaries. The proofs of the corollaries, observations and theorem in the remainder of this section, appear in the full paper.

Corollary 1. *Let* $\mathcal{R}$ *be a set of four open unit semicircles that cover the unit circle* $\mathbb{S}^1$. $\mathcal{R}$ *is minimal (i.e., for every open semicircle* $s \in \mathcal{R}$, $\mathcal{R} \setminus \{s\}$ *is not a covering set of* $\mathbb{S}^1$*) iff it consists of two antipodal pairs of open unit semicircles.*

Corollary 2. *Let* $\mathcal{S}$ *be a set of distinct open unit semicircles that covers* $\mathbb{S}^1$*; if* $|\mathcal{S}| \geq 5$, *then there exists* $\mathcal{R} \subset \mathcal{S}$, $|\mathcal{R}| = 3$ *and* $\mathcal{R}$ *covers* $\mathbb{S}^1$.

Generalizing Corollaries 1 and 2 to 3-space yields the following.

Corollary 3. *Let* $\mathcal{R}$ *be a set of six open unit hemispheres that cover the unit sphere* $\mathbb{S}^2$. $\mathcal{R}$ *is minimal iff it consist of three antipodal pairs of open unit hemispheres.*

Corollary 4. *Let* $\mathcal{S}$ *be a set of distinct open unit hemispheres that covers* $\mathbb{S}^2$*; if* $|\mathcal{S}| \geq 7$, *then there exists* $\mathcal{R} \subset \mathcal{S}$, $|\mathcal{R}| = 5$ *and* $\mathcal{R}$ *covers* $\mathbb{S}^2$.

When a facet f of the workpiece partially coincides with a facet of the fixture, the workpiece cannot translate in any direction that forms an acute angle with the (outer) normal to the plane containing f (without colliding with the fixture). This set of blocking directions comprises an open unit hemisphere denoted as

$h(f)$. Similarly, $H(\mathcal{F}) = \{h(f) \,|\, f \in \mathcal{F}\}$ denotes the mapping from a set of facets to the set of corresponding open unit hemispheres; see, e.g., [18]. Let $\mathcal{F}'$ denote the set of facets of the workpiece that are coincident with facets of the fixture in some fixed configuration. If the union of all blocking directions covers the unit sphere in that configuration, formally stated $\mathbb{S}^2 = \bigcup H(\mathcal{F}')$, then the workpiece cannot translate at all.

Let $\mathcal{F}$ denote the set of all facets of the fixture G. Let $\mathcal{F}_P$ denote the singleton that consists of the base facet of the palm of G, and let f_{b_i} and f_{t_i}, $1 \leq i \leq k$, denote the base facet of the body and the base facet of the fingertip, respectively, of the i-th finger of G, where k indicates the number of fingers. Let $\mathcal{F}_B = \{f_{b_i} \,|\, 1 \leq i \leq k\}$ and $\mathcal{F}_T = \{f_{t_i} \,|\, 1 \leq i \leq k\}$ denote the set of the base facets of the body parts of the fingers of G and the set of the base facets of the fingertip parts of the fingers of G, respectively. Let $\mathcal{F}_{PBT}$ denote the set of all base facets of the parts of G, that is $\mathcal{F}_{PBT} = \mathcal{F}_P \cup \mathcal{F}_B \cup \mathcal{F}_T$. Let $\mathcal{F}_{PB}$ denote the set of all base facets of the parts of G excluding the base facets of the fingertips, that is, $\mathcal{F}_{PB} = \mathcal{F}_P \cup \mathcal{F}_B$.

If the fixture resists any linear force applied on the workpiece while in the assembled state and there exists a collision free path (in the configuration space) between any separated configuration and the assembled configuration then our fixture is valid. We relax the second condition for practical reasons; instead of requiring a full path, we require a path of infinitesimal length. Formally we get:

Condition 1 : $\mathbb{S}^2 = \bigcup H(\mathcal{F}_{PBT})$.
Condition 2 : $\mathbb{S}^2 \neq \bigcup H(\mathcal{F}_{PB})$.

If the second condition holds, a serving state exists (assuming the flexibility of the joints cancels out the obstruction induced by the presence of the fingertips).

A candidate finger of an input polyhedron P is a valid finger of at least one possible fixture G of P.

Observation 1 *The number of candidate fingers of an input polyhedron P is linear in the number of vertices of P.*

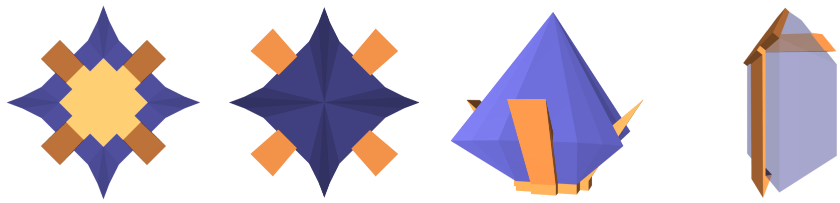
Theorem 1. *Every valid snapping fixture can be converted to a four-finger snapping fixture. Sometimes four fingers are necessary.*

Observation 2 *A single-finger fixture does not exist.*

A polyhedron that admits the lower bound is depicted in Figures 2a, 2b, and 2c. There there exists a polyhedron that has a snapping fixture that has only two fingers; see Figure 2d.

3 Algorithm

A snapping fixture G (of spreading degree two) is formally defined by a pair that consists of (i) an index i of a facet of P, and (ii) a set of pairs of indices $(j_1, \ell_1), (j_2, \ell_2), ...(j_k, \ell_k)$ of facets of P. The palm of G is the α_p-extrusion of the facet f_i. Each member pair of indices (j, ℓ) define a finger of G. The body and



(a) (b) (c) (d)

Fig. 2. (a), (b), (c) Different views of a polyhedron that has snapping fixtures with four fingers only and one of its four-finger fixtures. (d) A snapping fixture with two fingers.

fingertip of the finger are the α_b- and α_t-extrusion values of the facets f_j and f_ℓ, respectively.

Recently, we have introduced an algorithm that exhaustively searches through all valid snapping fixtures with 2, 3, or 4, fingers, of a given polyhedron and runs in $O(n^4)$ time [17]. Here, we introduce a much more parsimonious algorithm that uses a different method to generate 4-finger fixtures, yielding an algorithm that generates one fixture if exists with the minimal number of fingers and runs in $O(n^3)$ time.

Procedure 1 (MINIMALSNAPPINGFIXTURE(P)) The procedure accepts a polyhedron P as input and returns a fixture of P if exists with the minimal number of fingers; see Algorithm 1. The algorithm consists of two phases. In the first phase we compute a data structure M that associates palms and candidate fingers that extend from them. The second phase consists of three subphases in which we extract subsets of fingers of size, 2, 3, and 4, respectively, for each palm stored in M and examine whether the palm and the subset of fingers form a valid fixture. Once we strike one, we return it.

Procedure 2 (NEIGHBORS(f)) The procedure accepts a facet f of a polyhedron and returns all the neighboring facets of f.

Procedure 3 (SUBSETS($\mathcal{C}, k$)) The procedure accepts a set $\mathcal{C}$ and a positive integer k; it returns all subsets of $\mathcal{C}$ of cardinality k.

Procedure 4 (VALIDFIXTURE(F)) The procedure accepts a snapping fixture and determines whether it is a valid snapping fixture based on Conditions 1 and 2 defined in Section 2.5.

In each one of the subphases of the second phase we iterate over all facets of P and treat each facet as a potential base facet of the palm of a valid fixture (unless a fixture was found in a previous subphase). In the following, we narrow down the search space for fixtures with four fingers, once it has been established that our workpiece does not have a fixture with two or three fingers. Consider a

Algorithm 1 Minimal snapping fixture generation

Input: A polyhedron P with m facets $\{f_1, f_2, ..., f_m\}$.
Output: A snapping fixture G of P, if exists, with minimal number of fingers.

procedure $\textsc{minimalSnappingFixture}(P)$

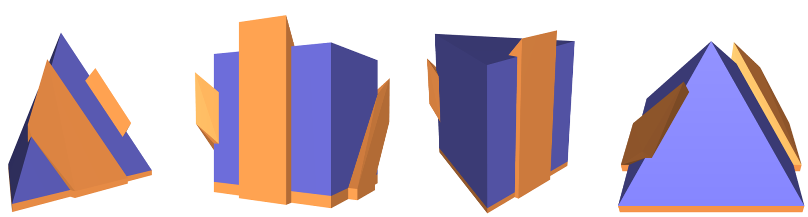
Phase 1
- **for** $i \leftarrow 1, m$ **do**
  - $M[i] \leftarrow \emptyset$
  - **for all** $j, f_j \in \textsc{neighbors}(f_i)$ **do**
    - **for all** $\ell, f_\ell \in \textsc{neighbors}(f_j)\ \&\ \ell \neq i$ **do**
      - $M[i] \leftarrow M[i] \cup \{(j, \ell)\}$

Subphase 2.1
- **for** $i \leftarrow 1, m$ **do**
  - **for all** $\mathcal{S}, \mathcal{S} \in \textsc{subsets}(M[i], 2)$ **do** // $|\mathcal{S}| = 2$
    - $F \leftarrow (f_i, \mathcal{S})$ // Define a fixture
    - **if** $\textsc{validFixture}(F)$ **then return** F

Subphase 2.2
- **for** $i \leftarrow 1, m$ **do**
  - **for all** $\mathcal{S}, \mathcal{S} \in \textsc{subsets}(M[i], 3)$ **do** // $|\mathcal{S}| = 3$
    - $F \leftarrow (f_i, \mathcal{S})$ // Define a fixture
    - **if** $\textsc{validFixture}(F)$ **then return** F

Subphase 2.3
- **for** $i \leftarrow 1, m$ **do**
  - $F \leftarrow \textsc{fourFingersFixture}(f_i, M[i])$
  - **if** $F \neq$ null **then return** F
- **return** null

polyhedron P that does have a valid fixture, say G (with an arbitrary number of fingers). There exists a subset $\mathcal{R} \subset H(\mathcal{F}_{BT})$, such that (i) $\mathcal{R}$ is a covering set of the closed hemisphere $\mathbb{S}^2 \setminus H(\mathcal{F}_P)$, and (ii) $|\mathcal{R}| \in \{3, 4, 5\}$. (This follows the same reasoning as in the proof of Theorem 1, which appears in [17].) The composition of R can be categorized into four cases listed below. We show that only one of theses cases, namely Case IV, must be considered when searching for a fixture with four fingers.



(a) (b) (c) (d)

Fig. 3. (a) A tetrahedron and a two-finger snapping fixture. (b) A cube and a three-finger snapping fixture. (c) A triangular prism and a two-finger snapping fixture. (d) A square pyramid and a two-finger snapping fixture.

Case I: $|\mathcal{R}| = 3$. The tetrahedron and the fixture depicted in Figure 3a demonstrate this case. At most three distinct fingers of G are needed; it implies that finding a fixture similar to G, but only with these three fingers, during the first or second subphases is guaranteed.

Case II: $|\mathcal{R}| = 5$. The tetrahedron and the fixture depicted in Figure 3b demonstrate this case. By Corollary 3, $\mathcal{R} \cup H(\mathcal{F}_P)$ consists of three antipodal pairs of open unit hemispheres. As $\mathcal{R} \cup H(\mathcal{F}_P)$ is a covering set of $\mathbb{S}^2$ and $|\mathcal{R} \cup H(\mathcal{F}_P)| = 6$, by Corollary 4, $H(\mathcal{F}_{PBT}) = \mathcal{R} \cup H(\mathcal{F}_P)$. It implies that the facets in $\mathcal{F}_{PBT}$ can be divided into three pairs of non-empty sets, such that each set is a collection of all facets with the same normal, and the two sets of every pair correspond to opposite normals, respectively. Without loss of generality, we assume that P does not have coplanar facets that are neighbors, because such facets can be merged. Next, observe that the facets in $\mathcal{F}_{PBT}$ must be parallelograms. Assume, for contradiction, that there exists a facet f that is not a parallelogram. It implies that f has at least three neighboring facets that are pairwise non-parallel, which implies that, together with $h(f)$, $\mathcal{R}$ contains at least four open hemispheres that are pairwise non-antipodal, a contradiction.

G must have at least one finger, say F_1, such that the normal to the base facet of its fingertip, say f_{t_1}, is opposite to the normal of the base facet of the palm f_p. Let f_{b_1} denote the base facet of the body of F_1. Consider the set $\mathcal{R}_1 = \mathcal{R} \setminus \{h(f_{b_1}), h(f_{t_1})\}$. Observe that $|\mathcal{R}_1| = 3$. Let $h(\bar{f_{b_1}})$ be the antipodal counterpart of $h(f_{b1})$. Consider the finger F_2, such that $\bar{f_{b_1}}$ is either the base facet, f_{b_2}, of the body of F_2 or the base facet, f_{t_2}, of the fingertip of F_2. Naturally, $h(\bar{f_{b_1}})$ is a member of $\mathcal{R}_1$. (i) If $\bar{f_{b_1}} = f_{t_2}$, then, since f_{b_2} is a neighbor of f_p and f_{t_2}, $h(f_{b_2})$ must be a member of $\mathcal{R}_1$ as well. Now, consider the set $\mathcal{R}_2 = \mathcal{R}_1 \setminus \{h(f_{b_2}), h(f_{t_2})\}$, and observe that $|\mathcal{R}_2| = 1$. (ii) If $\bar{f_{b_1}} = f_{b_2}$, then let $f_{t'}$ be one of the neighbors of f_{b_2} that is not parallel to f_p. Recall, that the facet f_{b_2} has four neighbors—two pairs of parallel facets. As f_{b_2} and f_{b_1} are parallel, $h(f_{t'})$ must be a member of $\mathcal{R}_1$ as well. If $f_{t'} \neq f_{t_2}$, replace the fingertip of F_2 with a fingertip, the base of which is $f_{t'}$. Now, consider the set $\mathcal{R}_2 = \mathcal{R}_1 \setminus \{h(f_{b_2}), h(f_{t'})\}$, and observe that $|\mathcal{R}_2| = 1$. It follows that there exists a third finger, say $F_3 \neq F_1, F_2$, such that either $h(f_{b_3}) \in \mathcal{R}_2$ or $h(f_{t_3}) \in \mathcal{R}_2$, where f_{b_3} and f_{t_3} are the base facets of the body and fingertip, respectively, of F_3, which obviates the need for further fingers. It implies that finding a valid fixture during the first or second subphases is guaranteed.

Case III: $|\mathcal{R}| = 4$ and there exists a facet $f \in \mathcal{R}$, such that $h(f)$ and $h(f_p)$ are antipodal. The triangular prism and the fixture depicted in Figure 3c demonstrate this case. As in the previous case, G must have at least one finger, say F_1, such that the normal to the base facet of its fingertip, say f_{t_1}, is opposite to the normal of the base facet of the palm f_p. Let f_{b_1} denote the base facet of the body of F_1. Consider the set $\mathcal{R}_1 = \mathcal{R} \setminus \{h(f_{b_1}), h(f_{t_1})\}$. Since $|\mathcal{R}_1| = 2$, at most two additional distinct fingers of G are needed; it implies that finding a fixture similar to G, but only with three fingers, during the first or second subphases is guaranteed.

Case IV: $|\mathcal{R}| = 4$ and $\mathcal{R}$ does not contain an open hemisphere, such that this hemisphere and $h(f_p)$ are antipodal. The square pyramid and the fixture depicted in Figure 3d demonstrate this case. Observe that the fixture in the figure has two fingers. However, sometimes four fingers are necessary as established by Theorem 1; see, e.g., Figure 2a. This is the only case we need to consider when searching for a fixture with four fingers. Notice, that in this case, the intersections of at least two open hemispheres in $\mathcal{R}$ with the great circle $\partial h(f_p)$ are pairwise antipodal open unit semicircles.

Procedure 5 (FOURFINGERSFIXTURES$(f, \mathcal{C})$) The procedure accepts a facet f of a potential palm and a set of pairs of facets, where each pair defines the base facets of the body and fingertip of a candidate finger, as input. It returns a valid fixture of P with four fingers, if there exists one, such that f is the base facet of its palm, and its configuration matches Case IV above. Let $\mathcal{C}'$ denote the set of unique facets in $\mathcal{C}$. Let $\bar{h} = \mathbb{S}^2 \setminus h(f)$ denote the closed hemisphere that must be covered by the open hemispheres $H(\mathcal{C}')$. The procedure first divides all the hemispheres in $H(\mathcal{C}')$ into equivalence classes, such that the intersections of all hemispheres in a class with the unit circle $C = \partial\bar{h}$ is a unique open semicircle. Let $s(\mathcal{E}) = x \cap C, x \in \mathcal{E}$ denote the unique open semicircle associated with the equivalence class $\mathcal{E}$. There is a canonical total order of hemispheres within each class: Let h_1 and h_2 be two hemispheres in some class; then $h_1 \prec h_2$ iff $h_1 \cap \bar{h} \subset h_2 \cap \bar{h}$. Then, the procedure identifies pairs of equivalence classes $(\mathcal{E}_1, \mathcal{E}_2)$, such that $s(\mathcal{E}_1)$ and $s(\mathcal{E}_2)$ are antipodal open semicircles. For each pair, the procedure traverses all other equivalence classes twice searching for two additional equivalence classes $\mathcal{E}_3$ and $\mathcal{E}_4$, such that the set $\{s(\mathcal{E}_1), s(\mathcal{E}_2), s(\mathcal{E}_3), s(\mathcal{E}_4)\}$ covers C. If it finds such four equivalence classes, it implies that there exists a valid fixture with four fingers F_1, F_2, F_3, F_4, such that the maximal hemisphere associated with $\mathcal{E}_i$ is either $h(f_{b_i})$ or $h(f_{g_i})$. In this case the procedure returns such a fixture.

The complexity of the algorithm is the accumulated complexities of Phase 1 and Subphases 2.1, 2.2, and 2.3. The efficiency (low running-time complexity) of Subphase 2.2 stems from an observation on the maximum number of possible candidates for this subphase, which in turn relies on the genus of the polyhedron, as we discuss next.

Lemma 1 (Genus of complete bipartite graphs [19]). *The genus of the complete bipartite graph, $k_{m,n}$, with m nodes in one side and n in the other, is $\lceil (m-2)(n-2)/4 \rceil$.*

Lemma 2. *Given an input polyhedron P of genus g. Let τ be a triplet of candidate fingers. Let $\mathcal{P}$ be the set of plams, such that all fingers in τ extend every palm in $\mathcal{P}$. Then, $|\mathcal{P}| \le 4 \cdot g + 2$.*

Proof. Let $\mathcal{A}$ be the set of three facets of P that correspond to the three base facets of the bodies of the fingers in τ. Let $\mathcal{B}$ be the set of facets of P that correspond to the base facets of the palms in $\mathcal{P}$. Let V, E, F denote the vertices, edges, and facets of P, respectively. Let $P^* = (V^*, E^*, F^*)$ be the dual graph of P, where each facet is represented as a node, and two nodes are connected by an arc if the corresponding two facets are neighbors. According to Euler characteristic, the genus of P^* is given by $1 - (|V^*| - |E^*| + |F^*|)/2$, which is equal to $1 - (|F| - |E| + |V|)/2 = g$. Consider the subgraph H of P^* that consists of the nodes that correspond to the facets in $\mathcal{A}$ and in $\mathcal{B}$. The genus of H is at most g. Since each facet in $\mathcal{A}$ and each facet in $\mathcal{B}$ are neighbors, H is a complete bipartite graph $k_{(3,|\mathcal{B}|)}$. By Lemma 1, the genus of H is $\lceil(3-2)(|\mathcal{B}|-2)/4\rceil = \lceil(|\mathcal{B}|-2)/4\rceil \leq g$. Hence, $|\mathcal{B}| \leq g \cdot 4 + 2$. □

Theorem 2. *Algorithm 1 runs in $O(n^3)$ time, where n is the number of vertices of the input polyhedron.*

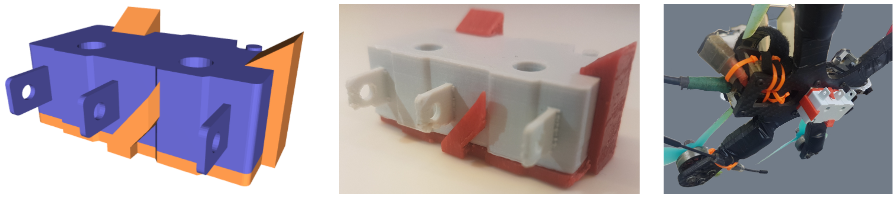
Proof. During the first phase we list all the potential palms, each palm together with all the fingers that can be connected to it. The overall number of potential fingers is twice the number of edges in the polytope; see Observation 1. Since The number of facets and the number of edges in a polytope with n vertices is linear in n, the number of palm-finger combinations created in Phase 1 is $O(n^2)$. The second phase dominates the time complexity. We examine each subphase separately. Recall, that a potential fixture passed to validFixture(F) (encoded by (f, S), where f denotes a facet and S denotes a set, the cardinality of which is fixed, i.e., 2, 3, or 4) has a fixed number of fingers. Therefore, every execution of the function consumes constant time. In the first subphase for every possible palm the function validFixture is invoked once per every subset of candidate fingers of size 2. As the number of candidate fingers is linear in n, the number of pairs of fingers is in $O(n^2)$. Thus, the total complexity of this subphase is $O(n \cdot n^2) = O(n^3)$. In the second subphase for every possible palm the function validFixture is invoked once per every subset of candidate fingers of size 3. By Lemma 2 and the assumption that the genus of the input polyhedron is constant, while iterating over all possible fixtures that have exactly three fingers, each triplet of fingers is considered a constant number of times. Therefore, the total time consumed processing potential fixtures of three fingers is bounded by $O(n^3)$. fourFingersFixtures($f, \mathcal{C}$) is invoked once for every facet in the input polyhedron. Building the equivalence classes and finding the maximum of each class takes $O(n)$ time. Matching maximal hemispheres of equivalence classes to form pairs of associated antipodal semicircles takes $O(n^2)$ time. Finally, examining every pair, traversing all other equivalence classes for each pair, also takes $O(n^2)$ time. Thus, the total complexity of this subphase is $O(n \cdot n^2) = O(n^3)$. The overall time complexity is thus $O(n^3)$. □

4 Two Applications

We present two applications that utilize our algorithm and its implementation.

4.1 Minimal Weight Fixtures

Generating lightweight fixtures that could be mounted on a UAV has been a major challenge ever since the first UAV was introduced. The desire for robust and efficient solutions to this problem rapidly scaled up during the last decade with the introduction of small drones, the weight of devices that can be mounted on which, is limited. Naturally, the device must be securely attached to the drone;



(a) (b) (c)

Fig. 4. (a) Synthetic micro switch sensor and a snapping fixture assembled. (b) A micro-switch sensor held by a fabricated snapping fixture. (c) A drone with the snapping fixture attached to it.

however, at the same time, the holding mechanism should weigh as little as possible. Figure 4 shows a fixture generated for a micro-switch sensor, a common sensor in the field of robotics and automation. Figure 4c shows the fabricated fixture (3D printed) permanently attached to a drone. It holds a micro-switch. While the micros-switch is firmly held during flight, it can be easily replaced.

4.2 Minimal Obscuring Fixtures

One of the objectives of jewelry making is to expose the gems mounted on a jewel, such as a ring, and reveal their allure. As with the minimal-weight fixture, the mounted gem must be securely attached to the jewel; however, the weight of the holding mechanism can be compromised. Here we seek to find a fixture that obscures the gem as little as possible, so that the gem surface is exposed as much as possible. Figure 5b shows a pendant with an integrated fixture synthesized by our generator. The fixture in Figure 5a is generated for an emerald cut; it reveals a surprising portion of the front facet of the stone.

(a) (b) (c)

Fig. 5. (a) an emerald cut—a common cut for precious stones. (b) a synthetic pendant with an integrated snapping fixture. (c) the fabricated pedant holding a precious stone.

4.3 3D Printing Considerations

We used various materials for generating snapping fixtures, such as, ABS, PLA, PETG, Nylon 12, and Sterling silver.[5] All generated fixtures properly snapped

[5] 3D printed wax and lost-wax where used to generate fixtures made of Sterling Silver.

and firmly held the workpieces. However, low quality prints (made of ABS, PLA, or PETG) occasionally broke after repeated or incautious uses. We noticed that increasing the infill density and orienting the prints such that the joint axes and the printing plate are not parallel increase the fixture durability. Also, we compensated for the limited precision of printers by scaling up the fixture to create a gap of up to 0.2mm between the fixture and the workpiece.

5 Experimental Results

The generator was developed in C++; it depends on the Polygon Mesh Processing package of CGAL [20]. Table 5 lists some of the workpieces we fed as input, and provides information about the generation of the corresponding snapping fixtures. The coordinates of the vertices of the input models were given in floating point numbers. The generator was executed on an *Intel Core i7-2720QM* CPU clocked at 2.2 GHz with 16 GB of RAM.

Table 1. Information related to snapping fixture generation of various workpieces. **Verts**, **Tris**, and **Fixts** stand for **Vertices**, **Triangles**, and **Fixtures**, respectively. The column entitled **Merged** indicates the number of facets after the merging of coplanar triangular facets. The last column indicates the number of fixtures that admit the minimal number of fingers.

Workpiece						**Fixture**		**# Fixts**
Name	**# Verts**	**# Edges**	**# Facets**		**Genus**	**# Min Fingers**	**Time (ms)**	**Min Fingers**
			Tris	**Merged**				
tetrahedron	4	6	4	4	0	2	3	36
dodecahedron[6]	20	30	36	12	0	2	15	50
emerald	34	96	64	25	0	2	39	8
square pyramid	5	8	6	5	0	2	4	24
micro switch	594	1,806	1,204	305	2	2	42,761	263,895
cube	8	18	12	6	0	3	20	216
octahedron	6	12	8	8	0	3	3	16
torus	32	64	32	10	1	3	307	2,760
4-finger	26	64	42	41	0	4	45	17
truncated cuboctahedron[6]	48	72	92	26	0	2	163	29
icosahedron	12	30	20	20	0	∞	22	0
8-base cylinder	16	42	28	10	0	2	44	106
28-base cylinder	56	162	108	30	0	2	984	4,396
48-base cylinder	96	282	188	50	0	2	4,672	24,456
68-base cylinder	136	402	268	70	0	2	13,008	71,892
88-base cylinder	176	522	348	90	0	2	27,233	159,124
108-base cylinder	216	642	428	110	0	2	50,122	297,956

Acknowledgement

The authors thank Shahar Shamai for his technical support, Omer Kafri and Raz Parnafes for fruitful discussions, and Shahar Deskalo for exploiting our work while crafting a jewel.

[6] Limited precision coordinates render the actual models non-regular.

References

1. H. H. Asada. Manufacturing robotics: Basic issues and challenges. *IFAC Proceedings Volumes*, 29(1):319–330, 1996.
2. M. Wilson. *Implementation of Robot Systems*. Elsevier, 1st edition, 2015.
3. D. Prattichizzo and J. C. Trinkle. *Handb. of Robotics*. Springer, 2nd edition, 2016.
4. X. Markenscoff, L. Ni, and C. H. Papadimitriou. The geometry of grasping. *Int. J. of Robotics Research*, 9:61–74, 1990.
5. B. Mishra, J. T. Schwartz, and M. Sharir. On the existence and synthesis of multifinger positive grips. *Algorithmica*, 2:541–558, 1987.
6. R. C. Brost and K. Y. Goldberg. A complete algorithm for synthesizing modular fixtures for polygonal parts. In *Proc. 1994 IEEE Int. Conf. on Robotics & Autom.*, volume 1, pages 535–542, 1994.
7. Y. Zhuang and K. Goldberg. On the existence of solutions in modular fixturing. *Int. J. of Robotics Research*, 15(6):646–656, 1996.
8. A. S. Wallack and J. F. Canny. Planning for modular and hybrid fixtures. *Algorithmica*, 19(1):40–60, 1997.
9. R. C. Brost and R. R. Peters. Automatic design of 3D fixtures and assembly pallets. *Int. J. of Robotics Research*, 17(12):1243–1281, 1998.
10. R. Wagner, Y. Zhuang, , and K. Goldberg. Fixturing faceted parts with seven modular struts. In *IEEE Int. Symp. on Assembly and Task Planning*, pages 133–139. IEEE Comput. Society Press, 1995.
11. Z. M. Bi and W. J. Zhang. Flexible fixture design and automation: Review, issues and future directions. *International Journal of Production Research*, 39(13):2867–2894, 2001.
12. A. Bicchi and V. Kumar. Robotic grasping and manipulation. In S. Nicosia, B. Siciliano, A. Bicchi, and P. Valigi, editors, *Ramsete. Lecture Notes in Control and Information Sciences*, volume 270. Springer, 2001.
13. K. Gopalakrishnan, K. Goldberg, G. M. Bone, M. J. Zaluzec, R. Koganti, R. Pearson, and P. A. Deneszczuk. Unilateral fixtures for sheet-metal parts with holes. *IEEE Trans. on Autom. Sci. and Eng.*, 1(2):110–120, 2004.
14. A. Sahbani, S. El-Khoury, and P. Bidaud. An overview of 3D object grasp synthesis algorithms. *Robotics and Autonomous Systems*, 60(3):326–336, 2012.
15. G. F. Liu, J. Xu, X. Wang, and Z. X. Li. On quality functions for grasp synthesis, fixture planning, and coordinated manipulation. *IEEE Trans. on Autom. Sci. and Eng.*, 1(2):146–162, 2004.
16. C. Rosales, J. M. Porta, and L. Ros. Grasp optimization under specific contact constraints. *IEEE Transactions on Robotics*, 29(3):746–757, 2013.
17. T. Tsabar, E. Fogel, and D. Halperin. Optimized synthesis of snapping fixtures. *The Comput. Research Repository*, abs/1909.05953, 2019.
18. P. Bose, D. Halperin, and S. Shamai. On the separation of a polyhedron from its single-part mold. In *Proc. 13th IEEE Int. Conf. on Autom. Sci. and Eng.*, pages 61–66. IEEE Comput. Society Press, 2017.
19. A. Bouchet. Orientable and nonorientable genus of the complete bipartite graph. *Journal of Combinatorial Theory, Series B*, 24(1):24–33, 1978.
20. S. Loriot, M. Rouxel-Labbé, J. Tournois, and I. O. Yaz. Polygon Mesh Processing. In CGAL *User and Reference Manual*. CGAL Editorial Board, 5.0.1 edition, 2020.