Optimized Synthesis of Snapping Fixtures

Thesis submitted in partial fulfillment of the requirements for the M.Sc. degree in the

Blavatnik School of Computer Science, Tel-Aviv University

by

Tom Tsabar

This work has been carried out under the supervision of
Prof. Dan Halperin

[February] 2022

Acknowledgement

I would like to 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. Special thanks for Efi Fogel for the technical support building the website, implementing the algorithms, editing and publishing our papers, printing 3D models, and providing deep insights about the nature of snapping fixtures. I would also like to thank Yoav Golan and Prof. Elon Rimon for reviewing our work, enriching our knowledge in the field of robotic grasping, and expanding our views on the use cases and the next steps implanting our theoretical results. Last I owe my gratitude to Prof. Dan Halperin for mentoring me from the very start through research, implementation, and publication of my work.

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 thesis 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, which 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; this algorithm runs in $O(n^4)$ time. We have deployed a public website that serves to construct snapping fixtures for given workpieces. The website design is described in the thesis as well. 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.

Contents

1 Introduction

A fixture is a device that holds a part in place. Constraining the movement of parts is a fundamental requirement in 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 closure* and *force closure* [3]. 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 thesis 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 separate, 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 common objective of the presented algorithms 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 fix-

ture 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

The arts of fixturing, grasping, and maintaining force closure have been studied by many researches over the years. This section provides background on central topics related to our problem.

Static fixtures

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 cases, 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] introduced the vise fixture and proposed an algorithm that automatically designs 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]. Subsequent works studied other types of fixtures and provided algorithms for computing them, for example, unilateral fixtures [12], which are used

to fix sheet-metal workpieces with holes.

Auto-generated holders oriented for 3D printing

Koyama et al [13] describe a complete system, called AutoConnect, for producing customizable, 3D-printable connectors attaching two physical objects together. They distinguish between two types of objects (to be attached). The first type is structured objects, objects in which their attachment area can be well approximated by a standard shape like a cylinder, a rectangular-prism, and a flat-plane. The second type of objects are free-form objects, which do not have standard shapes. They separately deal with the two types of objects. In order to hold a structured object, a mechanical holder is chosen from a database of six holders prepared in advance. The mechanical holder is customized by changing static parameters like width and thickness. In contrast, a specifically generated shell is created in order to hold a free-form object. The shell is created by extruding a set of connected facets of the input object. Their algorithm starts with a single facet of the input polyhedron, and accumulates neighboring facets until the so-called *holdability* criterion is met. The holdability criterion is an extension of the form-closure criterion discussed earlier.

The work in this thesis shares a similar goal with that of the AutoConnect approach: Generating a tailored 3D-printable holder for a given polyhedron. One major difference between the two approaches, is the method of solving the assembly problem, namely how the object and its holder are brought together. Notice that if the a fixture (or holder), blocks all moving directions, the workpiece cannot be assembled into the fixture. In our approach we address this problem by designing flexible snapping fixtures, which allow the fingers of the fixtures to bend during the assembly. The AutoConnect approach allows the user to choose one of two solutions. The first solution is leaving an exit direction and giving up on obtaining full form-closure. The second solution is dividing the holder into different pieces, connected together by snap-fit mechanisms. This difference leads to different fixture 3D design, and therefore different algorithms and correspondingly different analyses . Another difference is that our fixture is designed to be as lightweight or as less obscuring as possible, which is not necessarily a goal of AutoConnect when approaching free-formed objects.

Robotic grasping

While the thesis deals with fixturing, we see a similarity between methods of robot grasping and static fixturing and hence we briefly review several results in grasping. In grasp-synthesis with autonomous robotic *fingers*, a single robotic hand manipulated by motors, is used to grasp different workpieces; an overview of algorithms for such grasp synthesis can be found in [13]. In an early study, Hanafusa and Asada [14] proposed the use of flexible or elastic fingers in robotic grasping. They proposed a versatile robot hand with three fingers, which are driven by three individual motors through coil springs. Their goal was to handle two-dimensional objects dexterously by controlling finger forces. In later works [15, 16, 17] researchers suggested the use of robotic flexure-joints arms with semi-rigid fingers, where each finger is made from several joints connected by a hinge and a string. Each finger can be

bent separately by pulling its string with the relevant motor. Ma and Dollar even suggested, in a similar way to our work, the use of 3D printed fingers. They proposed a specific design[1] for such four-fingers robotic arm with flexure-joints [18].

A common dilemma for all the grasping and fixture design algorithms is defining and finding the optimal grasp. Several works, e.g., [19] and [20], discuss such objective functions and their optimization. Summary of the theoretical framework, analytical results, and open problems in robotic grasping can be found at [21]. In their recent book, Rimon and Burdick [22] comprehensively cover the design and analysis of multi-finger robot grasps. The book provides a detailed introduction to robot grasping principles, analyzes a variety of frictional and frictionless robot grasps, and discusses the grasping mechanisms. The book can provide inspiration for static fixtures as well. Few recent studies propose different approaches for the grasping problem. Pagoli *et al* [23] propose a soft robotic gripper with re-configurable fingers. The proposed fingers are made of silicone elastomer and are composed of three pneumatic chambers, which can be inflated independently. Therefore, the fingers can be bent, inflated and transformed in many ways achieving high flexibility in the grasping process. Another work [24] proposes roller based fingers. An active roller is embedded at the fingertip of each finger allowing easy manipulation of the workpiece mid-grip.

Deformable objects

We present some kind of *deformable* fixtures, which snap around solid workpieces. The studies of the dual problem, that is, grasping deformable objects with solid fingers, has some interesting insights that might help in accurate modeling of our deformable fixtures. Cui, Xiao and song [25] described a simulation for grasping of deformable convex objects. They modelled a realistic human hand and simulated the contact force and shape deformation during the process of grasping of an elastic object, achieving convincing simulated results. Later works [26, 27] proposed different strategies to grasp 3D deformable objects using multi-fingered robotic hands. They analyzed the contact model, described the object deformation, and characterized the grasp quality.

1.2 Our Results

We introduce 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 6.1a.[2] In our analysis in the sequel we assume that the genus is bounded by a constant. We introduce two algorithms. The first algorithm finds *all* valid fixtures for an input polyhderon P. This

[1] The 3D design and other details are available at https://www.eng.yale.edu/grablab/openhand/model_q.html

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

algorithm runs in $O(n^4)$ time. The second algorithm determines whether a fixture exists for an input polyhderon P, and if so, it constructs *one* in $O(n^3)$ time. We also provide an efficient and robust implementation of the former. 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. 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 works on semi-rigid one-part fixtures to compare to, but we provide benchmark statistics 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. An abridged summary of our work was published in 14th International Workshop on the Algorithmic Foundations of Robotics (WAFR) [28], and is available at https://link.springer.com/book/10.1007/978-3-030-66723-8.

1.3 Outline

The rest of this thesis is organized as follows. The geometric design of the snapping fixtures is described in Section 2. The analysis, theorems, and proofs regarding the geometric and combinatorial properties of the snapping fixtures are presented in Section 3. The synthesis algorithms are described in Section 4 along with the analysis of their complexity. Implementation details about the 3D design of the fixtures and about the website is provided in Section 5. Two applications are presented in Section 6. We report on experimental results in Section 7 and reflecting thoughts about future research in Section 8. Appendix featuring 3D printing consideration, and assortment of interesting workpieces and fixtures is provided in Sections A, and B.

1.4 Conventions

The following lists typical notations.

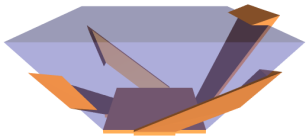
- $\mathbb{A}$, $\bar{\mathbb{A}}$—general open and closed unit semicircles, respectively
- $\mathbb{H}$, $\bar{\mathbb{H}}$—general open and closed unit hemispheres, respectively
- $\mathbb{P}^1$—the affinely extended real number line
- $\mathbb{P}^2$—a generalization of the affinely extended real number line to the plane
- $\mathbb{S}^1$—the unit circle
- $\mathbb{S}^2$—the unit sphere

- f—a facet
  - f_p—the base facet of the palm of a fixture
  - f_{b_i}—the base facet of the body of finger i
  - f_{t_i}—the base facet of the fingertip of finger i
- s, $\bar{s}$—instances of an open and a closed unit semicircle, respectively
- h, $\bar{h}$—instances of an open and a closed unit hemispheres, respectively
- $\mathcal{C}$, $\mathcal{E}$, $\mathcal{R}$, $\mathcal{S}$—sets
- $\mathcal{F}$—a set of facets
  - $\mathcal{F}_P$—the singleton that consists of the base facet of the palm of a fixture
  - $\mathcal{F}_B$—the set of the base facets of the bodies of the fingers of a fixture
  - $\mathcal{F}_T$—the set of the base facets of the fingertips of the fingers of a fixture
  - $\mathcal{F}_{PBT}$—the union of the above three sets
  - $\mathcal{F}^P$—the facets of a polyhedron P
- P,G—polyhedra, a workpiece and a snapping fixture, respectively
- $h(f)$—a mapping from a facet to the hemisphere that consists of the blocking directions induced by f
- $H(\mathcal{F})$—a mapping from a set of facets to the corresponding hemispheres

2 Design

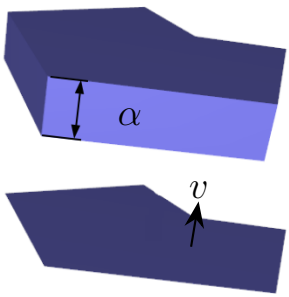
This chapter presents the structure of snapping fixtures and reviews considerations in their construction. The first section describes the structure of snapping fixtures, the next section refers to the assembly configurations of fixtures and workpieces, the third section addresses the restriction coming from the fixed distance between the palm facet and the fingertip facet, and the last section sets guiding lines for our fixture finding algorithms.

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 2.1 (**α**-extrusion of a polygon and base polygon of an α-extrusion)**.** *Let L denote a polygon, 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, 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.

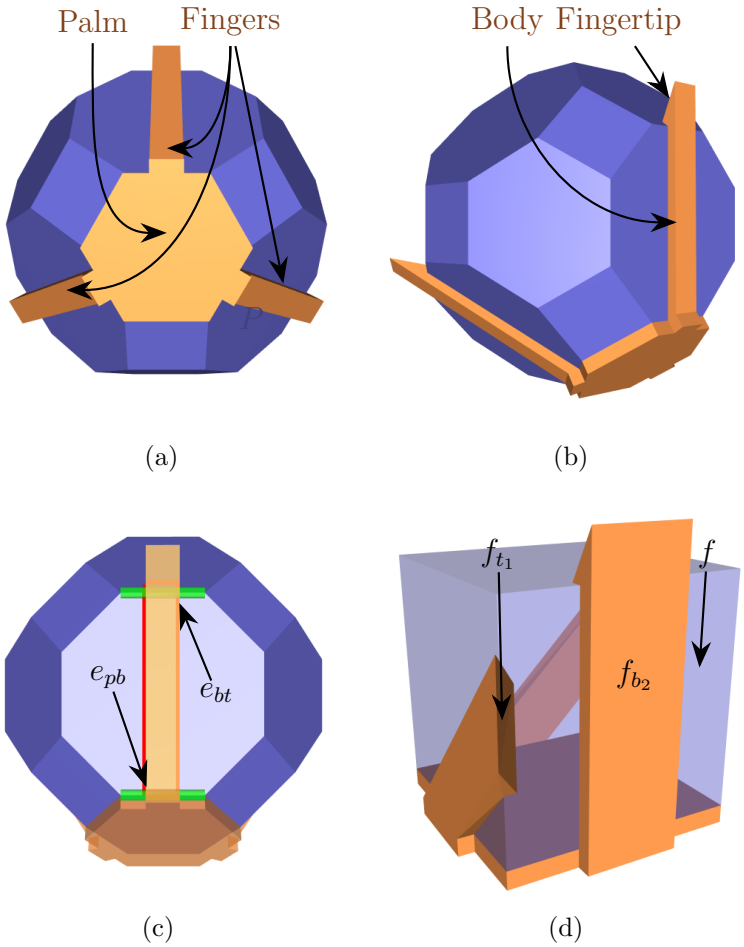
Let G denote a snapping fixture made of a palm, k fingers $F_1, F_2, \ldots, F_k$, and corresponding joints. The palm is an α_p-extrusion of a facet f_p of the workpiece P. (The various α values are discussed below.) Consider a specific finger $F = F_i$ of G. The body of F is defined by one of the neighboring 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.[1] They depend on the material and shape of the fixture. In our implementation they can be determined by the user.[2]

For a complete view of a workpiece and a snapping fixture consider Figure 2.1. 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 2.1d) 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 2.1(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

[1] Typically, these values are identical.

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

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 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.



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

Figure 2.1: (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.[3]

2.3 Spreading Degree

The *spreading degree* is the number of facets involved in the definition of a finger. In this thesis 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.

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 4, 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.

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

3

Combinatorial Analysis

The combinatorial analysis is organized as follows. In the first section of this chapter we convert the decision problem on validity of fixtures into a *covering-set* problem. The second section analyzes the properties of covering sets, setting the ground for the proofs and insights presented in the third section.

3.1 Valid Fixtures and Covering Sets

Definition 3.1 (open semicircle, open hemisphere)**.** *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 3.2 (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).

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., [29]. 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 move away from the fixture by translation.

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}, $1 \leqslant i \leqslant k$, 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 $\mathcal{F}_B = \{f_{b_i} \,|\, 1 \leqslant i \leqslant k\}$ and $\mathcal{F}_T = \{f_{t_i} \,|\, 1 \leqslant i \leqslant k\}$ 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 $\mathcal{F}_{PBT} = \mathcal{F}_P \bigcup \mathcal{F}_B \bigcup \mathcal{F}_T$. 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, $\mathcal{F}_{PB} = \mathcal{F}_P \bigcup \mathcal{F}_B$.

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 that the flexibility of the joints cancels out the obstruction induced by the presence of the fingertips.

3.2 Covering Set Properties

Theorem 3.3 (Helly's theorem [30]). *Let $\mathcal{S} = \{X_1, ..., X_n\}$ be a finite collection of convex subsets of $\mathbb{R}^d$, with $n > d$. If the intersection of every $d+1$ of these sets is nonempty, then the whole collection has a nonempty intersection; that is, $\bigcap_{j=1}^{n} X_j \neq \varnothing$.*

The contrapositive formulation of the theorem follows. If $\bigcap_{j=1}^{n} X_j = \varnothing$ then there exists a subset $\mathcal{R} = \{X_{i_1}, ..., X_{i_{d+1}}\} \subseteq \mathcal{S}$ such that $|\mathcal{R}| = d+1$ and $\bigcap_{j=1}^{d+1} X_{i_j} = \varnothing$. In the succeeding proofs we use the following corollary:

Corollary 3.4. *Let $\mathcal{S} = \{X_1, ..., X_n\}$ be a finite set of convex subsets of $\mathbb{R}^d$. If $\bigcup_{j=1}^{n} X_j = \mathbb{R}^d$ then there exists a subset $\mathcal{R} = \{X_{i_1}, ..., X_{i_{d+1}}\} \subseteq \mathcal{S}$ such that $|\mathcal{R}| = d+1$ and $\bigcup_{j=1}^{d+1} X_{i_j} = \mathbb{R}^d$.*

The corollary holds because the intersection of a set of subgroups of $\mathbb{R}^d$ is empty iff the union of their complement in $\mathbb{R}^d$ is $\mathbb{R}^d$.

The following four lemmas, namely, 3.5–3.8, are based on the analysis in [29].

Lemma 3.5. *Let $\mathcal{S}$ be a finite set of open unit semicircles. If $\mathcal{S}$ is a covering set of a closed unit semicircle $\bar{\mathbb{A}}$, then there exists $\mathcal{R} \subseteq \mathcal{S}$ such that $\mathcal{R}$ is a covering set of $\bar{\mathbb{A}}$ and $|\mathcal{R}| \in \{2, 3\}$.*

Proof. It is obvious that one open unit semicircle cannot cover a closed unit semicircle. Let $\mathbb{A}$ denote the interior of $\bar{\mathbb{A}}$. ($\mathbb{A}$ is an open unit semicircle.) There are two cases: (i) $\mathbb{A} \in \mathcal{S}$ and $\mathbb{A} \in \mathcal{R}$ for every covering set $\mathcal{R} \subseteq \mathcal{S}$ of $\bar{\mathbb{A}}$. It implies that every covering set

$\mathcal{R} \subseteq \mathcal{S}$ must contain two additional open semicircles that cover the two boundary points of $\bar{\mathbb{A}}$, respectively. These two semicircles together with $\mathbb{A}$ constitute a covering set of $\bar{\mathbb{A}}$ of size three. (ii) There exists a covering set $\mathcal{S}' \subseteq \mathcal{S}$, where $\mathbb{A} \notin \mathcal{S}'$. Let $\mathcal{S}'_{\bar{\mathbb{A}}} = \{s \bigcap \bar{\mathbb{A}} \,|\, s \in \mathcal{S}'\}$ be the set of intersections of the elements of $\mathcal{S}'$ and $\bar{\mathbb{A}}$. Let Π^1 denote the extended central projection that maps the closed semicircle $\bar{\mathbb{A}}$ to the *affinely extended real number line*,[1], $\Pi^1(p) = (x, w) : \bar{\mathbb{A}} \to \mathbb{P}^1$, where the points in $\mathbb{P}^1$ are represented in homogeneous coordinates (x, w). Notice that for every $s \in \mathcal{S}'_{\bar{\mathbb{A}}}$, s covers one of the boundary points of $\bar{\mathbb{A}}$; therefore, $\Pi^1(s)$ is an open ray covering either $(-1, 0)$ or $(+1, 0)$. $\mathcal{S}'_{\bar{\mathbb{A}}}$ covers $\bar{\mathbb{A}}$; therefore, the set of its images $\mathcal{S}'_{\Pi^1} = \{\Pi^1(s) \,|\, s \in \mathcal{S}'_{\bar{\mathbb{A}}}\}$ covers $\mathbb{P}^1$. By Helly's theorem, there exists a subset $\mathcal{R}'_{\Pi^1} \subseteq \mathcal{S}'_{\Pi^1}$ of size two that covers $\mathbb{R}^1$. Thus, the set of preimages of $\mathcal{R}'_{\Pi^1}$ covers $\bar{\mathbb{A}}$. □

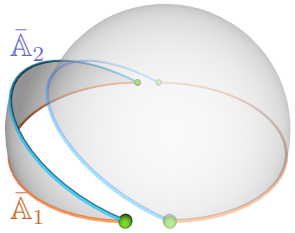
Lemma 3.6. *Let $\mathcal{S}$ be a finite set of open unit semicircles. If $\mathcal{S}$ is a covering set of the unit circle $\mathbb{S}^1$, then there exists $\mathcal{R} \subseteq \mathcal{S}$ such that $\mathcal{R}$ is a covering set of $\mathbb{S}^1$ and $|\mathcal{R}| \in \{3, 4\}$.*

Proof. Let $s \in \mathcal{S}$ be an arbitrary open unit semicircle in $\mathcal{S}$. The remaining elements $\mathcal{S}\backslash\{s\}$ of $\mathcal{S}$ must cover $\hat{s}$, the complement of s in the unit circle. Notice that $\hat{s}$ is a closed unit semicircle. By lemma 3.5, there exists $\mathcal{R}' \subseteq \mathcal{S}\backslash\{s\}$ that covers $\bar{s}$, and $|\mathcal{R}'| \in \{2, 3\}$. Thus, $\mathcal{R}' \bigcup \{s\}$ covers $\mathbb{S}^1$, and $|\mathcal{R}' \bigcup \{s\}| \in \{3, 4\}$. □

Lemma 3.7. *Let $\mathcal{S}$ be a finite set of open unit hemispheres. If $\mathcal{S}$ is a covering set of a closed unit hemisphere $\bar{\mathbb{H}}$, then there exists $\mathcal{R} \subseteq \mathcal{S}$, such that $\mathcal{R}$ is a covering set of $\bar{\mathbb{H}}$ and $|\mathcal{R}| \in \{3, 4, 5\}$.*

Proof. Let $\mathbb{H}$ denote the interior of $\bar{\mathbb{H}}$ ($\mathbb{H}$ is an open unit hemisphere) and $\partial\mathbb{H}$ denote the boundary of $\bar{\mathbb{H}}$ ($\partial\mathbb{H}$ is a great circle). Similar to the proof of Lemma 3.5, there are two cases: (i) $\mathbb{H} \in \mathcal{S}$ and $\mathbb{H} \in \mathcal{R}$ for every covering set $\mathcal{R} \subseteq \mathcal{S}$ of $\bar{\mathbb{H}}$. It implies that every covering set $\mathcal{R} \subseteq \mathcal{S}$ must contain additional open hemispheres that cover $\partial\mathbb{H}$. Let $\mathcal{S}_{\partial\mathbb{H}} = \{s \bigcap \partial\mathbb{H} \,|\, s \in \mathcal{S}\}$ be the set of intersections of the elements of $\mathcal{S}$ and $\partial\mathbb{H}$. Note that an intersection of a unit open hemisphere and a great circle is either empty or an open unit semicircle. Therefore $\mathcal{S}_{\partial\mathbb{H}}$ is a set of open unit semicircles lying on the same plane. By Lemma 3.6, there exists a covering set $\mathcal{R}_{\partial\mathbb{H}} \subset \mathcal{S}_{\partial\mathbb{H}}$ of $\partial\mathbb{H} = \mathbb{S}^1$, such that $|\mathcal{R}_{\partial\mathbb{H}}| \in \{3, 4\}$. This implies that there exists a covering set $\mathcal{R} \subseteq \mathcal{S}$ of $\bar{\mathbb{H}}$, such that $|\mathcal{R}| \in \{4, 5\}$.

(ii) There exists a covering set $\mathcal{S}' \subseteq \mathcal{S}$, where $\mathbb{H} \notin \mathcal{S}'$. Let $\mathcal{S}'_{\bar{\mathbb{H}}} = \{s \bigcap \bar{\mathbb{H}} \,|\, s \in \mathcal{S}'\}$ be the set of intersections of the elements of $\mathcal{S}'$ and $\bar{\mathbb{H}}$.



Let Π^2 denote the extended central projection that maps the closed hemisphere $\bar{\mathbb{H}}$ to an extended plane obtained by adjoining all signed slopes to $\mathbb{R}^2$ (a generalization of the affinely extended real number line, to the plane), $\Pi^2(p) = (x, y, w) : \bar{\mathbb{H}} \to \mathbb{P}^2$, where the points in $\mathbb{P}^2$ are represented in homogeneous coordinates (x, y, w).

Notice that every $s \in \mathcal{S}'_{\bar{\mathbb{H}}}$ is a semi-open spherical wedge; see the figure in the previous page. The wedge is bounded by two semicircles $\bar{\mathbb{A}}_1$ and $\bar{\mathbb{A}}_2$ (in the figure above), where $\bar{\mathbb{A}}_1$ lies in $\partial\mathbb{H}$. The intersection of $\bar{\mathbb{A}}_2$ and s is empty, and the intersection of $\bar{\mathbb{A}}_1$ and s is an

[1]The set $\mathbb{R}^1 \bigcup \{+\infty, -\infty\}$ is referred to as the affinely extended real number line.

open semicircle; therefore, $\Pi^2(s)$ is an open halfplane. $\mathcal{S}'_{\bar{\mathbb{H}}}$ covers $\bar{\mathbb{H}}$; therefore, the set of its images $\mathcal{S}'_{\Pi^2} = \{\Pi^2(s) \,|\, s \in \mathcal{S}'_{\bar{\mathbb{H}}}\}$ covers $\mathbb{P}^2$. By Helly's theorem, there exists a minimal subset $\mathcal{R}'_{\Pi^2} \subseteq \mathcal{S}'_{\Pi^2}$ of size at most three that covers $\mathbb{R}^2$. If $|\mathcal{R}'_{\Pi^2}| = 2$, that is, two open halfplanes, say h_1 and h_2 comprise $\mathcal{R}'_{\Pi^2}$, then they must be parallel: $h_1 : ax + by + c_1 > 0$ and $h_2 : ax + by + c_2 > 0$. In this case they do not cover the points $(-b, a)$ and $(b, -a)$ in $\mathbb{P}^2$. Thus, the pair of preimages of $\mathcal{R}'_{\Pi^2}$ covers $\bar{\mathbb{H}}$ except for two antipodal points. Covering these antipodal points requires two additional elements from $\mathcal{S}'_{\bar{\mathbb{H}}}$, which yields a covering set of size four. If $|\mathcal{R}'_{\Pi^2}| = 3$, then none of the halfplanes in $\mathcal{R}'_{\Pi^2}$ (which cover $\mathbb{R}^2$) are parallel, and they also cover $\mathbb{P}^2$. Thus, the set of preimages of $\mathcal{R}'_{\Pi^2}$ covers $\bar{\mathbb{H}}$, which yields a covering set of size three. □

Lemma 3.8. *Let $\mathcal{S}$ be a finite set of open unit hemispheres. If $\mathcal{S}$ is a covering set of the unit sphere $\mathbb{S}^2$, then there exists $\mathcal{R} \subseteq \mathcal{S}$ such that $\mathcal{R}$ is a covering set of $\mathbb{S}^2$ and $|\mathcal{R}| \in \{4, 5, 6\}$.*

Proof. Let $s \in \mathcal{S}$ be an arbitrary open unit hemisphere in $\mathcal{S}$. The remaining elements $\mathcal{S}\backslash\{s\}$ of $\mathcal{S}$ must cover $\hat{s}$ the complement of s in the unit sphere. Notice $\hat{s}$ is a closed unit hemisphere. By lemma 3.7, there exists $\mathcal{R}' \subseteq \mathcal{S}\backslash\{s\}$ that covers $\hat{s}$, and $|\mathcal{R}'| \in \{3, 4, 5\}$. Thus, $\mathcal{R}' \bigcup \{s\}$ covers $\mathbb{S}^2$, and $|\mathcal{R}' \bigcup \{s\}| \in \{4, 5, 6\}$. □

Corollary 3.9. *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 $s \in \mathcal{R}$, $\mathcal{R}\backslash\{s\}$ is not a covering set of $\mathbb{S}^1$) iff it consists of two antipodal pairs of open unit semicircles.*

Proof. ($\Rightarrow$) Assume, by contradiction, that $\mathcal{R}$ contains an open unit semicircle a, such that the interior of its complement is not in $\mathcal{R}$. Observe that the complement of a is a closed unit semicircle. This is exactly case (ii) in the proof of Lemma 3.5. Here, there exists a covering set $\mathcal{R}'$ of the closed unit semicircle, such that $|\mathcal{R}'| = 2$. It implies that $|\mathcal{R}|$ is at most three, a contradiction.

($\Leftarrow$) If $\mathcal{R}$ consist of two antipodal pairs of open unit hemispheres, then the removal of any one of the four hemispheres leaves one point on $\mathbb{S}^1$ uncovered. □

Corollary 3.10. *Let $\mathcal{S}$ be a set of distinct open unit semicircles that covers $\mathbb{S}^1$; if $|\mathcal{S}| \geqslant 5$, then there exists $\mathcal{R} \subset \mathcal{S}$, $|\mathcal{R}| = 3$ and $\mathcal{R}$ covers $\mathbb{S}^1$.*

Proof. Assume, for contradiction, that a subset $\mathcal{R} \subset \mathcal{S}$, $|\mathcal{R}| = 3$ that covers $\mathbb{S}^1$ does not exist. By Lemma 3.6, there exists a minimal subset $\mathcal{R}$ of $\mathcal{S}$ that covers $\mathbb{S}^1$ and $|\mathcal{R}| = 4$. By Corollary 3.9, $\mathcal{R}$ consists of two antipodal pairs of open unit semicircles. Let $\hat{a}$ denote the complement of the sole semicircle in $\mathcal{S}\backslash\mathcal{R}$. Observe that $\hat{a}$ is equivalent to the closed semicircle $\bar{\mathbb{A}}$, and that the interior of $\hat{a}$, is not in $\mathcal{R}$. This is exactly case (ii) in the proof of Lemma 3.5 where a semicircle $\bar{\mathbb{A}}$ is left to be covered and the interior of its complement is not in the covering set. Here, there exists a covering set $\mathcal{R}'$ of $\bar{\mathbb{A}}$, such that $|\mathcal{R}'| = 2$. It implies that $|\mathcal{R}| = 3$, a contradiction. □

Generalizing Corollaries 3.9 and 3.10 to 3-space yields the following.

Corollary 3.11. *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.*

Proof. ($\Rightarrow$) Assume, by contradiction, that $\mathcal{R}$ contains an open unit hemisphere a, such that the interior of its complement is not in $\mathcal{R}$. Observe that the complement of a is equivalent to $\mathbb{H}^2$. This is exactly case (ii) in the proof of Lemma 3.7. Here, there exists a covering set $\mathcal{R}'$ of $\mathbb{H}^2$, such that $|\mathcal{R}'| \in \{3, 4\}$. It implies that $|\mathcal{R}|$ is at most five, a contradiction.

($\Leftarrow$) $\mathcal{R}$ consists of three antipodal pairs of open unit hemispheres that cover $\mathbb{S}^2$. Arbitrarily pick one antipodal pair. There is a great circle c that it not covered by the pair. By corollary 3.9 two antipodal pairs of open unit semicircles are required to cover c; they must be the intersections of the remaining two antipodal pairs of open unit hemispheres, respectively. Thus, six open hemispheres are required in total. □

Corollary 3.12. *Let $\mathcal{S}$ be a set of distinct open unit hemispheres that covers $\mathbb{S}^2$; if $|\mathcal{S}| \geqslant 7$, then there exists $\mathcal{R} \subset \mathcal{S}$, $|\mathcal{R}| = 5$ and $\mathcal{R}$ covers $\mathbb{S}^2$.*

Proof. Assume, for contradiction, that a subset $\mathcal{R} \subset \mathcal{S}$, $|\mathcal{R}| = 5$ that covers $\mathbb{S}^2$ does not exist. By Lemma 3.6, there exists a minimal subset $\mathcal{R}$ of $\mathcal{S}$ that covers $\mathbb{S}^2$ and $|\mathcal{R}| = 6$. By Corollary 3.11, $\mathcal{S}$ consists of three antipodal pairs of open unit hemispheres. Let $\hat{h}$ denote the complement of the sole hemisphere in $\mathcal{S} \backslash \mathcal{R}$. Observe that $\hat{h}$ is equivalent to $\bar{\mathbb{H}}$, and that the interior of $\hat{h}$, is not in $\mathcal{R}$. This, again, is exactly case (ii) in the proof of Lemma 3.7. Here, there exists a covering set $\mathcal{R}'$ of $\bar{\mathbb{H}}$, such that $|\mathcal{R}'| \in \{3, 4\}$. It implies that $|\mathcal{R}| \leqslant 5$, a contradiction. □

3.3 Insights on Snapping Fixtures

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

Definition 3.13 (genus of a polyhedron)**.** *The genus of a polyhedron P is the genus of the graph induced by the vertices and edges of all the facets of P [31].*

Observation 3.14. *The number of candidate fingers of an input polyhedron P is linear in the number of vertices of P.*

Proof. Let e be an edge of P and let f_e and f'_e be the two faces incident to e. Two fingers can be built on e. The base facet of the body and the base facet of the tip of one finger coincides with f_e and f'_e, respectively. In order to construct the other finger, the roles of these facets exchange; that is, the base facet of the body and the base facet of the fingertip coincides with f'_e and f_e, respectively. Every candidate finger is built on a single edge. Thus, the number of candidate fingers is at most $2|E|$. From Euler's formula we know that the number of edges in a polyhedron of genus zero with n vertices is at most $3n - 6$. Thus, the number of candidate fingers is at most $6n - 12$. □

Theorem 3.15. *Every valid snapping fixture can be converted into a four-finger snapping fixture. Sometimes four fingers are necessary.*

Proof. Consider a polyhedron P. Let G be a valid fixture of P, and assume that G has more than four fingers. We show that it is possible to construct a valid snapping fixture of P that has (i) the same palm as G, and (ii) four fingers that are a subset of the fingers of G. Consider the closed hemisphere $\bar{\mathbb{H}} = \mathbb{S}^2 \backslash H(\mathcal{F}_P)$. By Condition 1 defined in Section 3.1, $\mathbb{S}^2 = \bigcup H(\mathcal{F}_{PBT})$. We get that $\bar{\mathbb{H}} \subseteq \bigcup H(\mathcal{F}_{BT})$. In other words, $H(\mathcal{F}_{BT})$ is a covering set of $\bar{\mathbb{H}}$. By Lemma 3.7, there exists a subset $\mathcal{R} \subset H(\mathcal{F}_{BT})$, such that (i) $\mathcal{R}$ is a covering set, and (ii) $|\mathcal{R}| \in \{3, 4, 5\}$. We prove separately for $|\mathcal{R}| \in \{3, 4\}$ and $|\mathcal{R}| = 5$.

If $|\mathcal{R}| \in \{3, 4\}$, there exist $i \in \{3, 4\}$ hemispheres that correspond to i base facets of i bodies or fingertips, respectively, of at most four fingers, which we choose as the fingers of G'.

If $|\mathcal{R}| = 5$, then $\mathcal{R}$ contains an open hemisphere $\mathbb{H}_t$, such that $\mathbb{H}_t = h(f_t)$ and the base facet of the palm and f_t are parallel.[2] In a polyhedron, two parallel facets cannot be neighbors; thus, f_t must be the base facet of a fingertip of some finger F. Let f_b denote the base facet of the body of the finger F and set $\mathbb{H}_b = h(f_b)$. Observe, that $\mathcal{R}_1 = \mathcal{R} \backslash \{\mathbb{H}_t\}$ must be a covering set of the unit circle $\partial\mathbb{H}_t$, and $|\mathcal{R}_1| = 4$. Observe that $\partial\mathbb{H}_b \neq \partial\mathbb{H}_t$; thus, $\mathcal{R}_2 = \mathcal{R}_1 \backslash \{\mathbb{H}_b\}$ is a covering set of a closed semicircle $\bar{\mathbb{A}}$ and $|\mathcal{R}_2| = 3$. Following a deduction similar to the above, there exist three hemispheres that correspond to three base facets of three bodies or fingertips, respectively, of at most three fingers, which we choose as the fingers of G' in addition to F.

A polyhedron that admits the lower bound is depicted in Figure 3.1. Proving that a snapping fixture for this polyhedron with less then four fingers does not exists is done using our generator (see Section 7). We exhaustively searched the configurations space and did not find a valid snapping fixture with less than four fingers. □

Observation 3.16. *A single-finger fixture does not exist.*

Proof. Let G be a fixture with only one finger. Then, $|H(\mathcal{F}_{PBT})| = 3$. However, by Lemma 3.8 the minimum size of a covering set of $\mathbb{S}^2$ is four. □

A polyhedron that admits the lower bound is depicted in Figures 3.1a, 3.1b, and 3.1c. There exists a polyhedron that has a snapping fixture that has only two fingers; see Figure 3.1d.

[2] Similar conditions are described in the proof of Lemma 3.7.

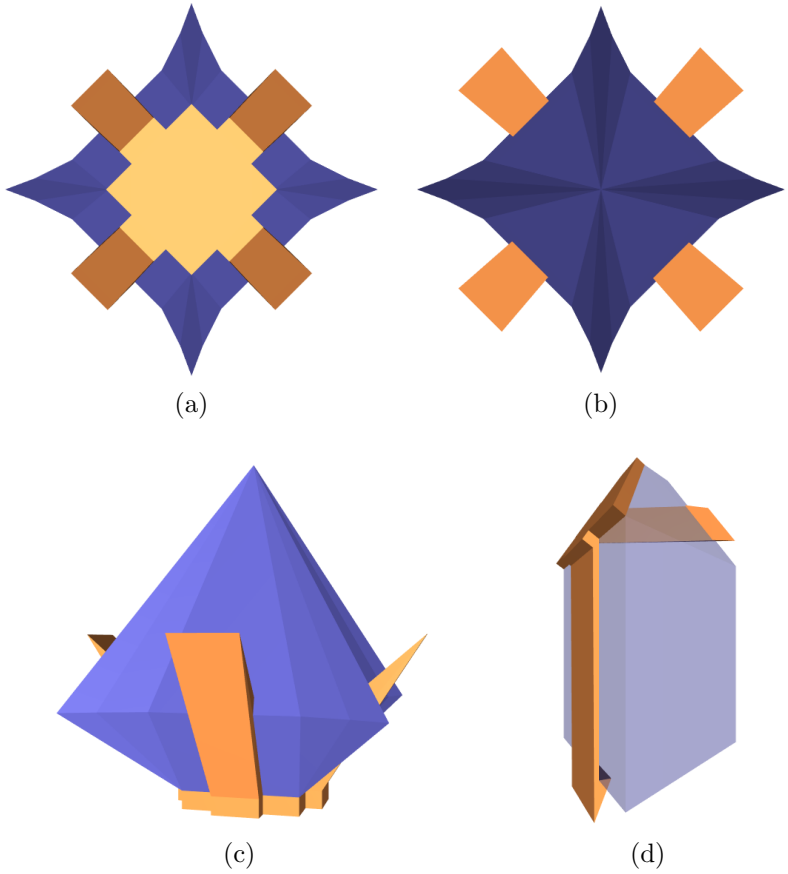


Figure 3.1: (a), (b), (c) Different views of a polyhedron that has snapping fixtures with four fingers only and one of its four-finger fixtures. (d) A snapping fixture with two fingers.

4
Algorithms

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

We introduce two algorithms. The first algorithm exhaustively searches and outputs all valid snapping fixtures with 2, 3, or 4, fingers, of a given polyhedron and runs in $O(n^4)$ time. The second algorithm is more parsimonious; it uses a different method to generate 4-finger fixtures, producing one fixture if exists with the minimal number of fingers and runs in $O(n^3)$ time.

4.1 Simple Algorithm

Procedure 1. (**minimalSnappingFixtures**(P)) The procedure accepts a polyhedron P as input and returns all fixtures of P 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 accumulate it to the list of valid fixtures.

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

Algorithm 1 Minimal snapping fixtures generation

Input: A polyhedron P with m facets $\{f_1, f_2, ..., f_m\}$.
Output: All minimal snapping fixtures of P.

1: **procedure** MINIMALSNAPPINGFIXTURES(P)
2: $fixtures \leftarrow \varnothing$
3: **for** $i \leftarrow 1, m$ **do** } Phase 1
4: $M[i] \leftarrow \varnothing$
5: **for all** $j, f_j \in$ NEIGHBORS(f_i) **do**
6: **for all** $\ell, f_\ell \in$ NEIGHBORS(f_j) $\& \ell \neq i$ **do**
7: $M[i] \leftarrow M[i] \bigcup \{(j, \ell)\}$
8: **for** $i \leftarrow 1, m$ **do** } Subphase 2.1
9: **for all** $\mathcal{S}, \mathcal{S} \in$ SUBSETS($M[i], 2$) **do** // $|\mathcal{S}| = 2$
10: $F \leftarrow (f_i, \mathcal{S})$ // Define a fixture
11: **if** VALIDFIXTURE(F) **then**
12: $fixtures \leftarrow fixtures \bigcup \{F\}$
13: **if** $fixtures \neq \varnothing$ **then return** $fixtures$
14: **for** $i \leftarrow 1, m$ **do** } Subphase 2.2
15: **for all** $\mathcal{S}, \mathcal{S} \in$ SUBSETS($M[i], 3$) **do** // $|\mathcal{S}| = 3$
16: $F \leftarrow (f_i, \mathcal{S})$ // Define a fixture
17: **if** VALIDFIXTURE(F) **then**
18: $fixtures \leftarrow fixtures \bigcup \{F\}$
19: **if** $fixtures \neq \varnothing$ **then return** $fixtures$
20: **for** $i \leftarrow 1, m$ **do** } Subphase 2.3
21: **for all** $\mathcal{S}, \mathcal{S} \in$ SUBSETS($M[i], 4$) **do** // $|\mathcal{S}| = 4$
22: $F \leftarrow (f_i, \mathcal{S})$ // Define a fixture
23: **if** VALIDFIXTURE(F) **then**
24: $fixtures \leftarrow fixtures \bigcup \{F\}$
25: **return** $fixtures$

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 3.

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. We accumulate all the valid fixtures during the process. 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 4.1 (Genus of complete bipartite graphs [32])**.** *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 4.2. *Given an input polyhedron P of genus g. Let τ be a triplet of candidate fingers. Let $\mathcal{P}$ be the set of palms, such that all fingers in τ extend every palm in $\mathcal{P}$. Then, $|\mathcal{P}| \leqslant 4 \cdot g + 2$.*

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 4.1, the genus of H is $\lceil (3-2)(|\mathcal{B}| - 2)/4 \rceil = \lceil (|\mathcal{B}| - 2)/4 \rceil \leqslant g$. Hence, $|\mathcal{B}| \leqslant g \cdot 4 + 2$. □

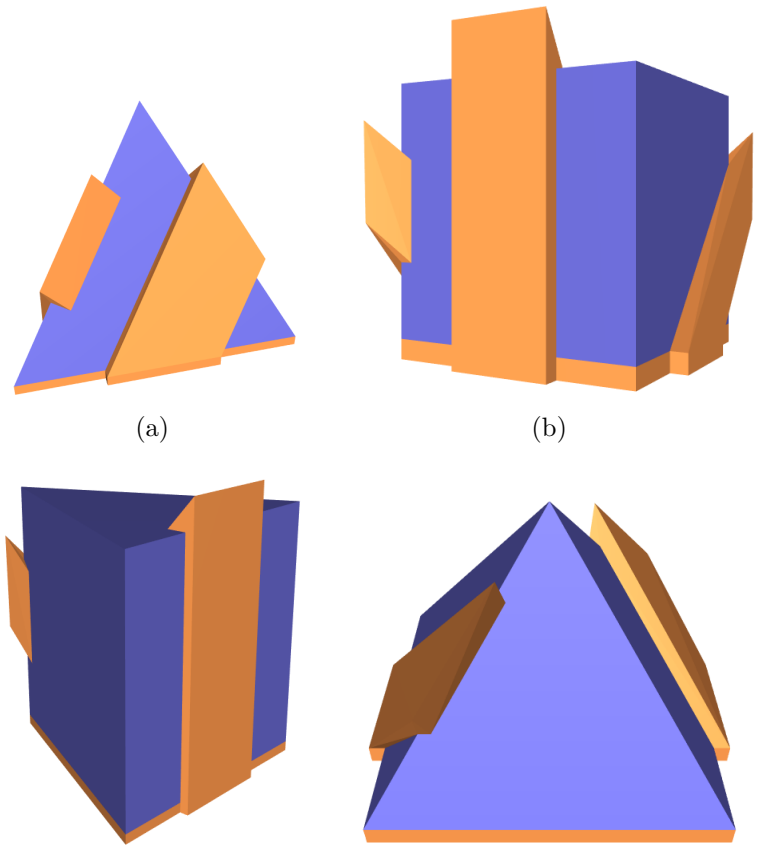
Theorem 4.3. *Algorithm 1 runs in $O(n^4)$ time, where n is the number of vertices of the input polyhedron.*

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 3.14. 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 4.2 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 $O(n^3)$. In the third subphase for every possible palm the function validFixture is invoked once per every subset of candidate fingers of size 4. For every candidate quadruple that is considered for a palm in the third subphase, each triplet of fingers in the quadruple was considered for the same palm in the second subphase. Each triplet of fingers is considered a constant number of times, therefore each quadruple of fingers is considered a constant number of times. Therefore, the total time consumed processing potential fixtures of four fingers is bounded by $O(n^4)$. The overall time complexity is thus $O(n^4)$. □

4.2 Improved Algorithm

The following procedure finds a four-fingers fixture if exists, assuming fixtures with two or three fingers do not exist. The procedure runs in $O(n^3)$ time.

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 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 $\mathbb{S}^2 \backslash H(\mathcal{F}_P)$, and (ii) $|\mathcal{R}| \in \{3, 4, 5\}$. (This follows the same reasoning as in the proof of Theorem 3.15) 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)

Figure 4.1: (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 4.1a 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 4.1b demonstrate this case. By Corollary 3.11, $\mathcal{R} \bigcup H(\mathcal{F}_P)$ consists of three antipodal pairs of open unit

hemispheres. As $\mathcal{R} \bigcup H(\mathcal{F}_P)$ is a covering set of $\mathbb{S}^2$ and $|\mathcal{R} \bigcup H(\mathcal{F}_P)| = 6$, by Corollary 3.12, $H(\mathcal{F}_{PBT}) = \mathcal{R} \bigcup H(\mathcal{F}_P)$. 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 $\mathcal{R}_1 = \mathcal{R} \backslash \{h(f_{b_1}), h(f_{t_1})\}$. 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 $\mathcal{R}_2 = \mathcal{R}_1 \backslash \{h(f_{b_2}), h(f_{t_2})\}$, 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 $\mathcal{R}_2 = \mathcal{R}_1 \backslash \{h(f_{b_2}), h(f_{t'})\}$, 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 4.1c 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 $\mathcal{R}_1 = \mathcal{R} \backslash \{h(f_{b_1}), h(f_{t_1})\}$. 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 4.1d demonstrate this case. Observe that the fixture in the figure has two fingers. However, sometimes four fingers are necessary as established by Theorem 3.15; see, e.g., Figure 3.1a. 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 $\bar{h} = \mathbb{S}^2 \backslash h(f)$ 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 $s(\mathcal{E}) = x \bigcap C, x \in \mathcal{E}$ 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 $h_1 \bigcap \bar{h} \subset h_2 \bigcap \bar{h}$. 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.

Algorithm 2 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 MINIMALSNAPPINGFIXTURE(P)
    for i ← 1, m do                                                    } Phase 1
        M[i] ← ∅
        for all j, f_j ∈ NEIGHBORS(f_i) do
            for all ℓ, f_ℓ ∈ NEIGHBORS(f_j) & ℓ ≠ i do
                M[i] ← M[i] ⋃ {(j, ℓ)}
    for i ← 1, m do                                                    } Subphase 2.1
        for all S, S ∈ SUBSETS(M[i], 2) do          // |S| = 2
            F ← (f_i, S)                            // Define a fixture
            if VALIDFIXTURE(F) then return F
    for i ← 1, m do                                                    } Subphase 2.2
        for all S, S ∈ SUBSETS(M[i], 3) do          // |S| = 3
            F ← (f_i, S)                            // Define a fixture
            if VALIDFIXTURE(F) then return F
    for i ← 1, m do                                                    } Subphase 2.3
        F ← FOURFINGERSFIXTURE(f_i, M[i])
        if F ≠ null then return F
    return null
```

In each one of the first two 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, this time, stopping after the first valid fixture found.

Theorem 4.4. *Algorithm 2 runs in $O(n^3)$ time, where n is the number of vertices of the input polyhedron.*

Proof. The time complexity analysis of phase 1, subphase 2.1 and subphase 2.2 is the same as in Theorem 4.3. The sole difference in the time analysis is the substitution of the original subphase 2.3 with the FOURFINGERSFIXTURES$(f, \mathcal{C})$ procedure. 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 cumulative time complexity is thus $O(n^3)$. □

5 Implementation Details

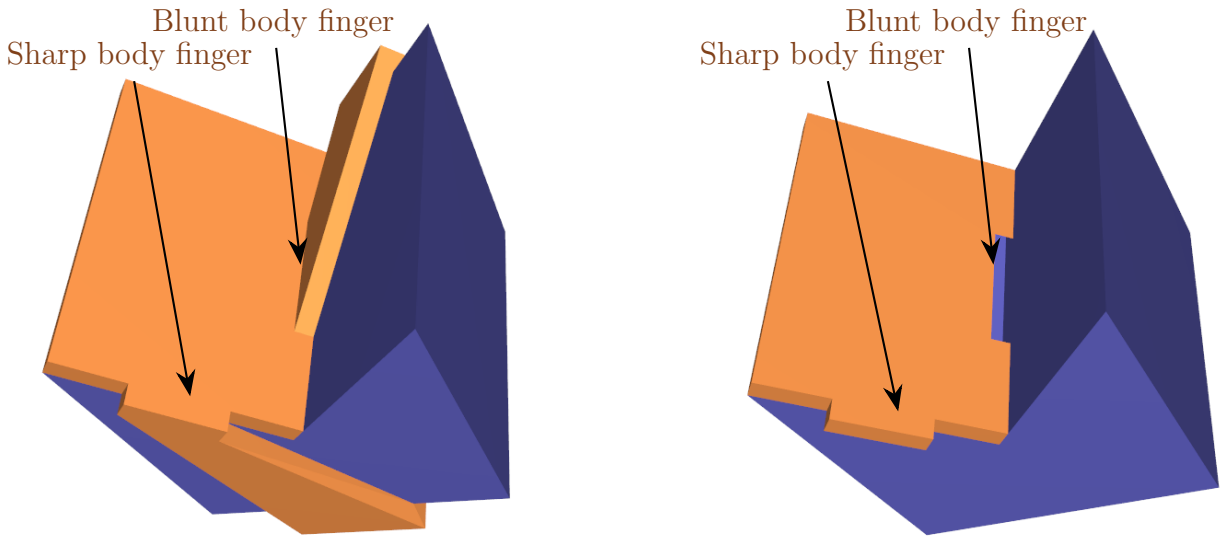
This chapter provides additional details on the 3D modeling process of the fixtures, the website and user-interface considerations. Our code is based on two open-source libraries that helped us manipulating geometrical figures and generate 3D models of the fixtures. The first library is CGAL (available at https://www.cgal.org/); it provides easy access to efficient and reliable geometric algorithms and data structure. The second library is SGAL (available at https://bitbucket.org/efifogel/sgal); it provides merging of coplanar facets, 3D formats conversions and additional utilities. We also used the code base of covering set validation by Shahar Shamai. The website was developed using standard libraries and tools, such as, Flask (https://flask.palletsprojects.com/en/2.0.x/), Nginx (https://www.nginx.com/), Docker (https://www.docker.com/), three.js (https://threejs.org/), supervisord (http://supervisord.org/), gunicorn (https://gunicorn.org/).

5.1 3D Modeling

We describe the 3D modeling process that outputs a ready-for-printing fixture from a set of palm, fingers, and fingertips facets. This process can be separated into several steps. The first step accepts four parameters from the user, namely palm thickness, finger thickness, fingertip depth and fingertip thickness in order to generate custom fixtures that suits the user requirements. Before reading the next section it is recommended to have a brief look at Section 2.1.

Palm Generation

The palm of the fixture is bounded by several facets. We start with the base facet, the second is an extrusion of the first facet. The extrusion is built from the same vertices of the first facet, translated in the direction of the normal to the palm facet; its length is *palm thickness*. Except from preserving the shape of the palm facet, the second facet also supports connection of fingers to the palm; this connection is described later. In order to reduce the volume of the fingers and avoid collisions, the finger body's facet is defined to be the convex hull of *part* of the segment e_{pb} and of *part* of segment e_{bt}. In our implementation we shrank each of the segments symmetrically to the length of $0.4 \cdot L$, where L is the segment original length; see Figure 2.1(c). The first distinction is between connections for blunt body fingers and sharp body fingers; see Figure 5.1. For sharp body fingers the palm extends e_{pb}. For blunt body fingers the palm creates a *rebate* in e_{pb} that the finger can connect to without colliding with the workpiece. Note that the extrusion or rebate are at length of $finger\ thickness$; see Figure 5.1(b).



(a) (b)

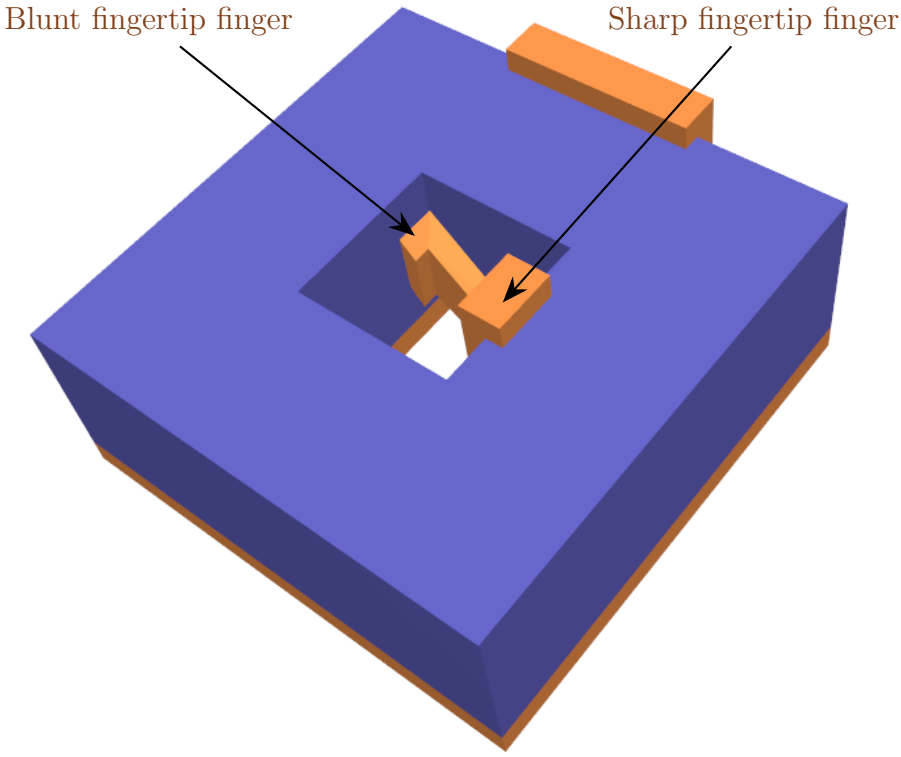
Figure 5.1: (a) Workpiece and fixture with two fingers, one has blunt body and one sharp body. (b) Same workpiece and fixture, but the fixture has no fingers.

The last step of generating the palm is filling the space between the two facets with the palm side facets.

Finger and Fingertip Generation

Every finger is extruded from its related connection (extrusion or rebate), at the palm, until intersected with the fingertip facet's plane. Then the fingertip is extruded from the finger in the direction normal to fingertip facet's plane and in length of *fingertip thickness* . A second distinction is made between blunt fingertip fingers and sharp fingertip fingers. While sharp fingertip bend towards the workpiece such that the inner facet of the finger snaps onto

the workpiece, blunt fingertips bend towards the workpiece such that the outer facet of the finger pushes the workpiece outwards; See Figure 5.2a.



(a)

Figure 5.2: Blunt and Sharp fingertip fingers and their way of grasping the workpiece

Either the inner fingertip facet or outer fingertip facet, the one which is in direct contact with the workpiece, is set to be in length of $fingertip\ depth$ and while the other facet is set to match in length. The last part is completing the side facets for both the finger body and fingertip.

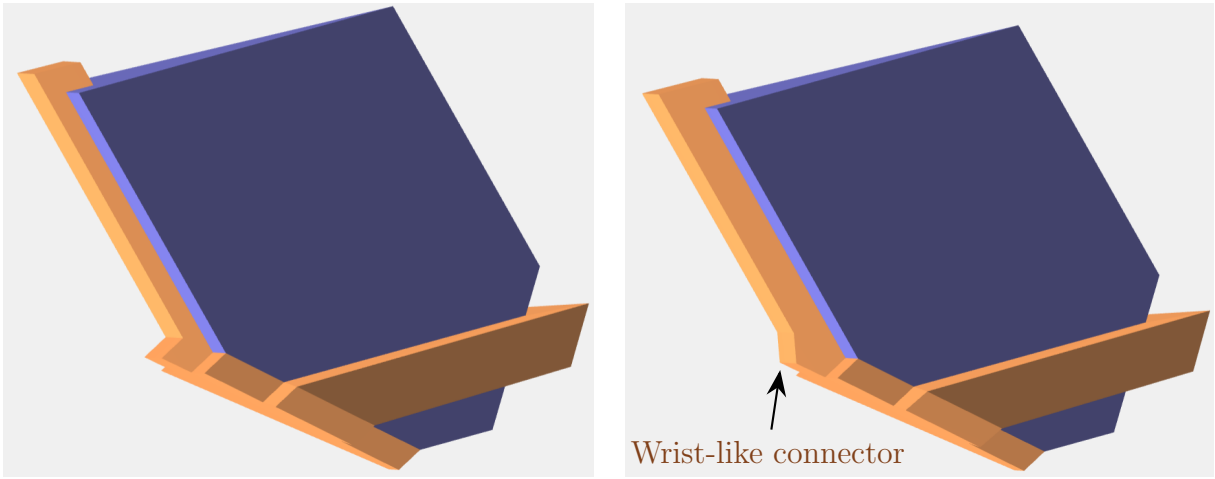
Blunt Body Finger Size

While experimenting and generating fixtures for different workpieces with different parameters we noticed the following constraint. Blunt body fingers' size was defined by *finger thickness*. However, when *finger thickness* is larger than the palm facet's dimensions, the rebate extends outside of the palm and the fixture becomes undefined. Therefore, if the rebate extends outside of the palm an error message appears alerting the user to decrease the *finger thickness*.

Angle Based Normalization

During our experiments we noticed that some fingers are thicker than other. However, all the fingers should have the same *finger thickness* parameter. The reason for this gap is illustrated in Figure 5.3(a), that the real thickness of a finger is not determined only by the

length of the connector l_{conn}, but by $l_{conn} \cdot \sin(\theta_{pb})$ where θ_{pb} is the angle between the palm facet and the body facet. In order to be fully convinced it is helpful to think about the edge case where the body and palm facets are near coplanar and the real thickness of the finger tends to zero. In order to normalize the thickness of all the fingers we multiply the length of the inner connector by $1/\sin(\theta_{pb})$. This addition creates a joint or wrist-like connector, as seen in Figure 5.3(b).



(a) (b)

Figure 5.3: (a) Before normalizing the palm connector. (b) After normalizing the palm connector.

Format Support and Coplanar Facets

Working with VRML and OFF file formats to represent the polyhedron and the fixture was straight-forward. In order to support the common STL format we had to apply few enhancements. The first change was in the data encoding. Both VRML and OFF formats are textual formats, but STL has a binary encoding that does not work well when sending the data over http textual protocol. The solution was to encode the data before sending it, using Base64 encoding scheme.[1] The major problem with the STL format is the lack of combinatorial information. When STL is generated every facet is triangulated and added to the collection of triangles; there is no data available regrading each triangle original facet. In the parsing process the code builds a polyhedron from the STL file, trying to merge coplanar triangles into a single facet. The limited precision of STL makes it hard to recognize which triangles are coplanar and originate from the same facet. Our solution is setting a threshold—only triangles with normal vectors n_1, n_2 that are ϵ close to each other are considered coplanar ($|cos^{-1}(n_1 \cdot n_2/|n_1||n_2|) - \pi| < \epsilon$). Another issue we had to address is handling vertices of non-triangular facets that are not coplanar. Fear of complex "fixing" algorithms, which might destroy the fragile details of the polyhedron surface while trying to avoid non-coplanar facets, led us to preserve the polyhedron with non-coplanar facets. The advantage of this approach is that we can generate fixtures without worrying that they

[1] Additional details about Base64 encoding can be found at https://en.wikipedia.org/wiki/Base64

might not fit the original workpiece. The disadvantage is that the algorithm loses its ability to determine if two facets are parallel, which is required for an accurate analysis of covering sets. Note that only when the service is fed with a model in the STL format the algorithm loses its ability to distinguish between parallel and almost parallel facets, when using VRML and OFF formats the user can exploit the accurate version of the algorithm.

Special Cases Topology

In some special cases, the set of facets that defines the fixture is valid (that is, conditions 1 and 2, from Section 3 hold), but the described modeling process outputs an unprintable 3D model. The first example is of a fixture that can be 3D modeled, but the generated fixture *skips* the *body* facet of the finger. As seen in Figure 5.4 below, the relative position of the palm, body and fingertip facets outputs a finger that does not cover the body facet; therefore, the fixture will not counter a translation in the direction of the outer normal of the body's facet. The next examples, illustrated in Figure 5.5, present a valid fixture, and two facet

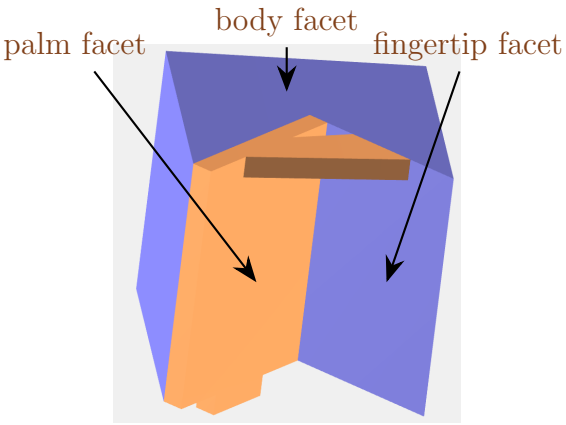
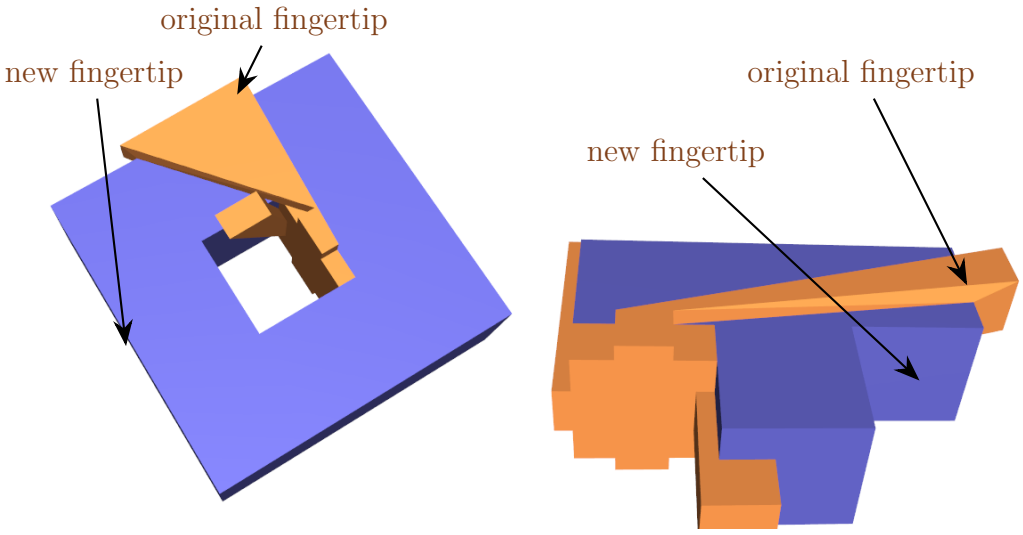


Figure 5.4: A non-convex workpiece with its degenerated fixture.

(original and new), such that if we replace the original facet in the fixture with the new one, we get an unprintable 3D model that cannot be rendered. Since the 3D model cannot be rendered to an image, we demonstrate these fixtures by showing a valid fixture, and how it can be changed into a fixture that cannot be rendered. The first example in Figure 5.5 (a) illustrates a fixture of a boxed torus. When selecting the new fingertip facet as proposed in the figure, the new finger covers the *hole* of the torus, and therefore collides with any finger that uses the facets of the *hole*, creating self-intersecting fixture. The second example, depicted in Figure 5.5 (b) illustrates a fixture of a T-shaped block. When selecting the new fingertip facet as proposed in the figure, the new finger's body segment (e_{pb}) and fingertip segment (e_{bt}) are coincident to a single line. Therefore, the finger body facet, defined as the convex hull of the two segments, is a degenerated quadrilateral of 0-width.

All the above examples can be solved by a more accurate design of the fingers, based on the exact shapes and relations between the facets and by using collision detection to avoid collisions between fingers. Although we believe it is possible to define this accurate design, it is out of scope of our current work.



(a) (b)

Figure 5.5: (a) A boxed torus with two related fixtures. (b) A T-shaped block with a related fixture. All figures present a small change in the facet selection that results in a fixture that can not be built the standard way described in the thesis so far.

5.2 Website

Instant Output

The implemented algorithm runs in $O(n^4)$ time in the worst-case and in $O(n^3)$ time in most cases; therefore, waiting for the algorithm to end might be tedious. Luckily, the user does not need to wait for the process of examining all possible fixtures to complete. The execution stops after the first valid fixture is found, then, it generates the fixture and saves its current configurations. The service allows the user to change the palm or the fingers selections. In most cases the response is immediate, which creates a pleasant user experience. The primary exception is when the user inputs a workpiece that does not have a valid fixture at all. In this case the procedure must examine all possible fixtures in order to conclude that a valid fixture does not exists.

Palm Selection

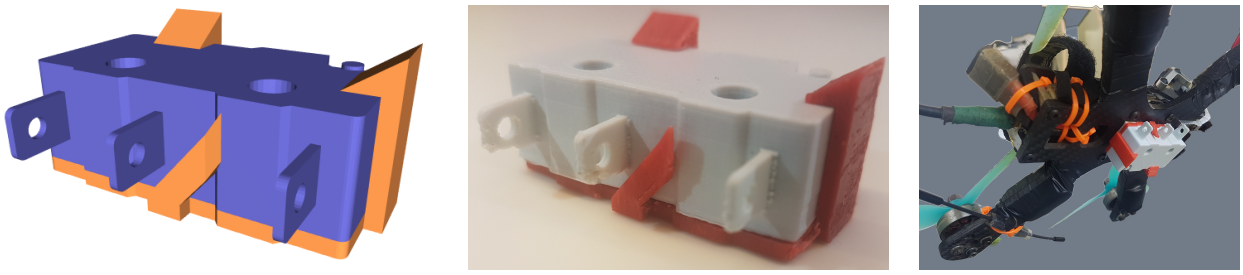
In order to improve the user experience, we supply an option to choose the palm base facet by clicking on it with the mouse. The option is enabled by picking the selection tool from the palm menu, and clicking on the desired facet. In order to implement this option, the algorithm merges coplanar (or near coplanar) facets, triangulates them, and generates a mapping in the process from every resulting triangle facet to its original root facet. The mapping is forwarded to the client, that renders two instances of the workpiece. First, the colored workpiece is rendered onto a visible buffer. Second, the workpiece is rendered onto an invisible buffer used for picking. When a triangle facet f is projected onto the invisible

buffer, the *colors* of the rendered fragments are assigned with the id of the root facet of f. When a facet is selected the client pulls the id of the selected facet from the invisible buffer and sends it back to the server, which generates an appropriate fixture based on the selected facet.

6 Two Applications

We present two applications that utilize our algorithms and their implementation.

6.1 Minimal Weight Fixtures



(a) (b) (c)

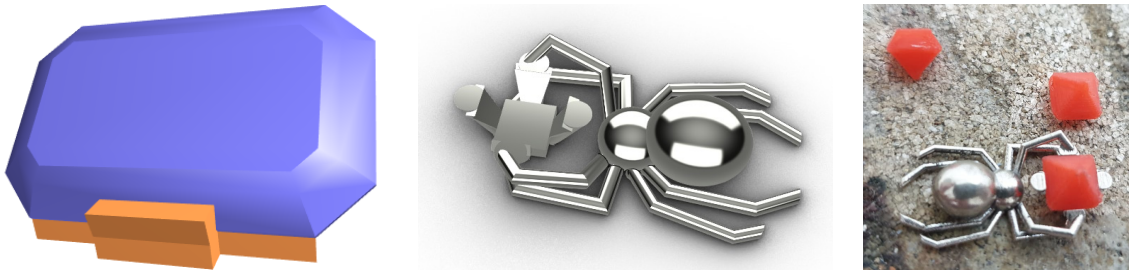
Figure 6.1: (a) Synthetic micro-switch sensor and a snapping fixture assembled. (b) A micro-switch sensor held by a fabricated snapping fixture. (c) A drone with the snapping fixture attached to it.

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; however, at the same time, the holding mechanism should weigh as little as possible. Figure 6.1 shows a fixture generated for a micro-switch sensor, a common sensor in the field of robotics and automation. Figure 6.1c 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.

6.2 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 6.2b shows a pendant with an integrated fixture synthesized by our generator. The fixture in Figure 6.2a is generated for an emerald cut; it reveals a surprising portion of the front facets of the stone.



(a) (b) (c)

Figure 6.2: (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.

7

Experimental Results

The generator was developed in C++; it depends on the *Polygon Mesh Processing* package of CGAL [33]. Table 7.1 (on the next page) 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.

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

Table 7.1: 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[1]	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[1]	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

8

Limitations and Future Research

8.1 Form Closure

Our generator synthesizes fixtures that do not necessarily prevent angular motion. Such fixtures are rarely obtained. Nevertheless, the figure to the right depicts a workpiece and a snapping fixture (synthesized by our generator), such that the workpiece can escape the assembled configuration using torque. However, other snapping fixture of this workpiece that guarantee form-closure of the workpiece do exist (and offered by our generator). Devising efficient synthesis algorithms for guaranteeing form closure is left for future research.

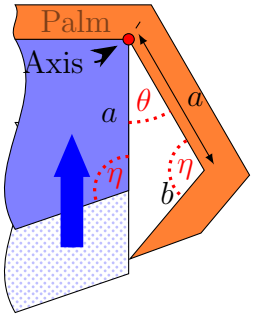
8.2 Spreading Degree

Increasing the *spreading degree* (see Section 2.3) will enable the synthesis of fixtures for a larger range of workpieces. Future research could result with (i) a classification of polyhedra according to the minimal spreading degree required for their snapping fixtures, and (ii) algorithms for synthesis of fixtures with a larger fixed spreading degree or even unlimited.

8.3 Joint Flexibility

The flexibility of the *joints* is an important consideration in the design. In order to construct a snapping fixture, the joint that connects the body of a finger to the palm, and the joint that connects the fingertip of a finger to its body must allow the rotation of the respective parts

about the respective axes when force is applied. Some of the subtleties of this flexibility are discussed below. For simplicity we move the discussion to the plane, where our workpiece and snapping fixture are polygons.



Let's focus on one finger. Consider the configuration where the finger is about to snap. Assume, for further simplicity, that the joint that connects the body and the fingertip of the finger is rigid, and consider only the joint that connects the finger with the palm, as depicted in the figure to the right. This configuration occurs a split second before the assembly reaches the assembled state when translated, starting at the serving configuration. Let θ denote the angle between the finger and the workpiece. Note that in the assembled configuration θ equals 0 for all fingers. Let θ_c denote the joint threshold angle, that is, the maximum bending angle the finger can tolerate without breaking. The threshold angle of every joint depends on the material and thickness of the region around the joint. θ is an angle of a triangle with one edge lying on the fixture's finger body inner facet and another edge lying on the fixture's fingertip inner facet. Let a and b denote the lengths of these edges, respectively, and let η be the angle between them. The finger will break when $\theta > \theta_c$. Applying the law of sines, we get $b = \frac{a \sin\theta}{\sin(\pi-\theta-\eta)} = \frac{a \sin\theta}{\sin(\theta+\eta)}$, which implies a maximal value $b \leqslant \frac{\min(a) \sin\theta}{\sin(\theta+\eta)}$. On the other hand, the characteristics of the material of the finger determine the minimal value of b that guarantees a secured grasp of the workpiece by the fingertip. The construction of a fixture G is feasible, only if selecting a proper value b for every finger of G is possible. We remark that the full analysis in space is more involved, and for now our generator does not take into account material properties such as flexibility.

8.4 Gripping Strength

Another consideration in the fixture design is the gripping strength. The gripping strength of a finger is based on the angle between the palm and the body of the finger and on the angle between the body of the finger and the fingertip of the finger. The gripping strength is in opposite relation with these angles; that is, the smaller each one of these angles is the stronger the gripping is. While our generator currently does not take in account strength considerations, it could be used as a criterion in ranking valid snapping fixtures of a given workpiece.

A

3D Printing Considerations

We used various materials for generating snapping fixtures, such as, ABS, PLA, PETG, Nylon 12, and Sterling silver.[1] All generated fixtures properly snapped and firmly held the workpieces. However, low quality prints (made of ABS, PLA, or PETG) occasionally broke after repeated or incautious use. 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.

Finger thickness

The finger-thickness parameter affects the flexibility of the finger. Our experiments show that narrow fingers are more flexible and wide fingers are more solid. If the finger is too flexible, the grip of the workpiece becomes weaker and the finger might slip off the workpiece. If the finger is to wide, the finger might not be flexible enough to stretch, bend, and snap onto the workpiece. We recommend matching the thickness of the fingers to the flexibility of the material, for example, more flexible materials (Nylon 12) should be used with wider fingers.

Printing orientation

3D objects can be printed in different orientations on the printing platform. Our 3D printer (as well as many others), fills a 2D layer of the object and then continues to the next layer. The slicer program itself chooses how to fill each 2D layer, but can be configured. The

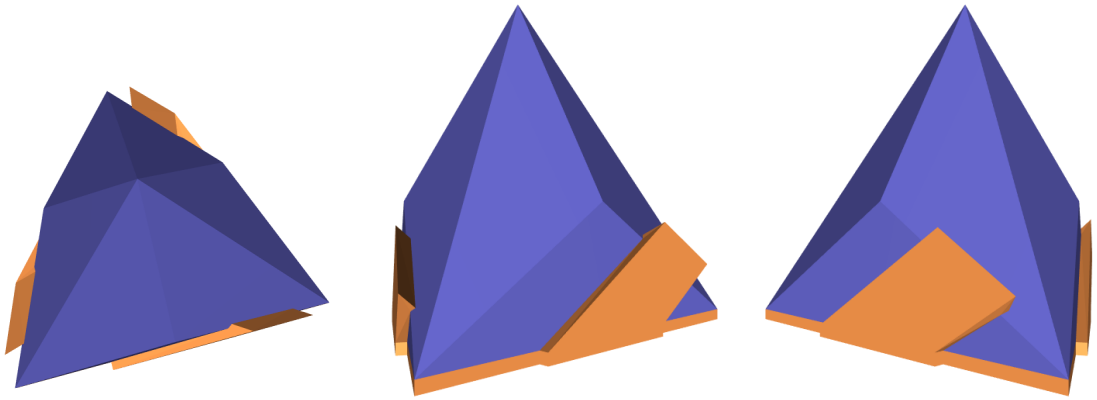
[1] 3D printed wax and lost-wax were used to generate fixtures made of Sterling silver.

connection between the layers is weaker than the forces connecting each string of filament. Therefore, we recommend 3D printing the fixture such that the fingers span as few layers as possible; this way stronger fingers are printed, less likely to break.

B

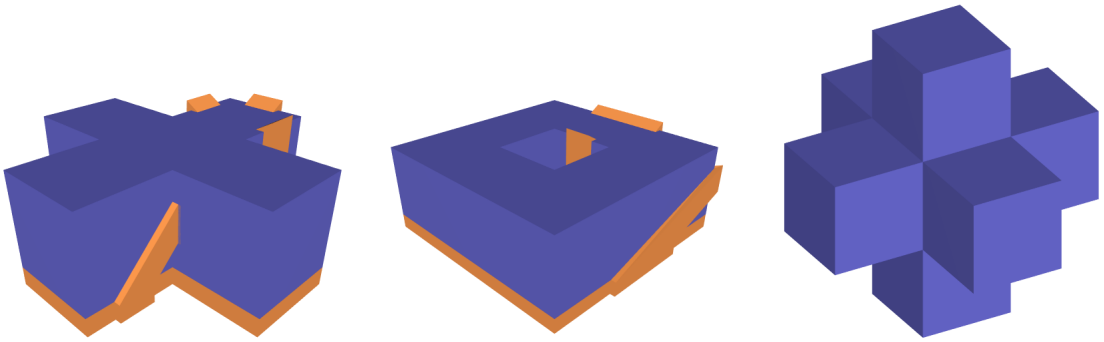
Assortment of Interesting Workpieces and Fixtures

Figure B.1 depicts (three views of) a polyhedron P and a fixture with three fingers that snaps onto P. It demonstrates case I in Section 4. Here, we fix the base facet of the palm. It holds that for every possible fixture of P with the fixed palm in the figures $|\mathcal{R}| = 3$. To construct the polyhedron P in the figure we start with a regular tetrahedron (such as the one depicted in Figure 4.1a), fix the bottom facet, subdivide each one of the remaining three facets into three identical triangles, and slightly translate the newly introduced vertex in the direction of the outer normal to the original facet, ensuring that the dihedral angle between the bottom facet and its neighbor remains acute.



(a) (b) (c)

Figure B.1: Three different views of a polyhedron with 10 facets and a three-finger snapping fixture.

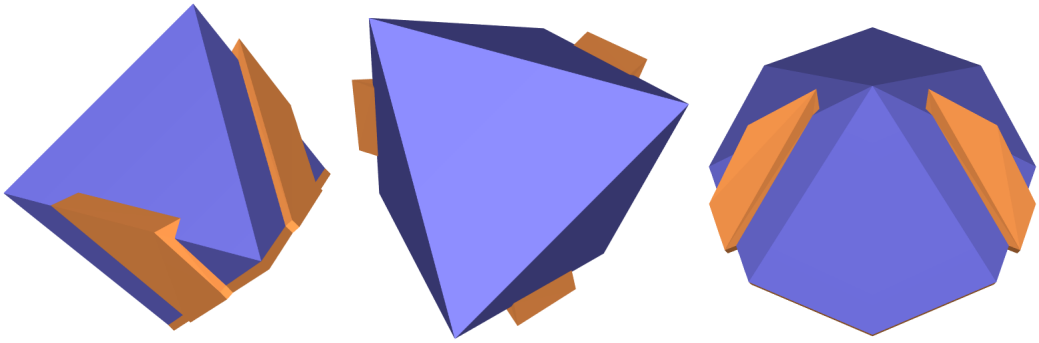


(a) (b) (c)

Figure B.2: (a),(b) Two polyhedra and their snapping fixtures, respectively. (c) A polyhedron that does not have a valid snapping fixture.

Figures B.2a and B.2b depict two polyhedra, P_1 and P_2, and their snapping fixtures, respectively. They demonstrate case I in Section 4. The number of facets of each polyhedron is larger than six; however, it holds that for every possible fixture of P_i, $|\mathcal{R}| = 5$, where $\mathcal{R} \subset H(\mathcal{F}_{BT})$ and $\mathcal{R}$ is a covering set of the closed hemisphere $\mathbb{S}^2 \backslash H(\mathcal{F}_P)$.

There exists a polyhedron P that does not have a valid fixture and the cardinality of the minimal covering set of $H(\mathcal{F}^P)$ is 6, where $\mathcal{F}^P$ is the set of all facets of the polyhedron P; see the Figure B.2c.



(a) (b) (c)

Figure B.3: (a),(b) Two different views of an octahedron and a three-finger snapping fixture. (c) An octagonal-pyramid and a two-finger snapping fixture.

Figures B.3a and B.3b depict an octahedron and a snapping fixture with three fingers, which is the minimum in this case. Figure B.3c depict an octagonal pyramid and a snapping fixture with two fingers.

Bibliography

[1] Haruhiko H. Asada. Manufacturing robotics: Basic issues and challenges. *IFAC Proceedings Volumes*, 29(1):319–330, 1996.

[2] Mike Wilson. *Implementation of Robot Systems*. Elsevier, 1st edition, 2015.

[3] Domenico Prattichizzo and Jeffrey C. Trinkle. Grasping. In Bruno Siciliano and Oussama Khatib, editors, *Handb. of Robotics*, chapter 38, pages 671–700. Springer, 2nd edition, 2016.

[4] Xanthippi Markenscoff, Lugan Ni, and Christos H. Papadimitriou. The geometry of grasping. *Int. J. of Robotics Research*, 9:61–74, 1990.

[5] Bud Mishra, Jacob Theodore Schwartz, and Micha Sharir. On the existence and synthesis of multifinger positive grips. *Algorithmica*, 2:541–558, 1987.

[6] Randy C. Brost and Kenneth Y. Goldberg. A complete algorithm for synthesizing modular fixtures for polygonal parts. In *Proc. 1994 IEEE Int. Conference on Robotics & Autom.*, volume 1, pages 535–542, 1994.

[7] Yan Zhuang and Ken Goldberg. On the existence of solutions in modular fixturing. *Int. J. of Robotics Research*, 15(6):646–656, 1996.

[8] Aaron S. Wallack and John F. Canny. Planning for modular and hybrid fixtures. *Algorithmica*, 19(1):40–60, 1997.

[9] Randy C. Brost and Ralph R. Peters. Automatic design of 3D fixtures and assembly pallets. *Int. J. of Robotics Research*, 17(12):1243–1281, 1998.

[10] Richard Wagner, Yan Zhuang, , and Ken 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] Zhuming 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] Kanakasabapathi Gopalakrishnan, Ken Goldberg, Gary M. Bone, Matthew J. Zaluzec, Rama Koganti, Rich Pearson, and Patricia A. Deneszczuk. Unilateral fixtures for sheet-metal parts with holes. *IEEE Trans. on Autom. Sci. and Eng.*, 1(2):110–120, 2004.

[13] Anis Sahbani, Sahar El-Khoury, and Philippe Bidaud. An overview of 3D object grasp synthesis algorithms. *Robotics and Autonomous Systems*, 60(3):326–336, 2012.

[14] Hideo Hanafusa and Haruhiko H. Asada. A robot hand with elastic fingers and its application to assembly process. *IFAC Proceedings Volumes*, 10(11):127–138, 1977. IFAC International Symposium on Information-Control Problems in Manufacturing Technology, Tokyo, Japan, 17-20 October.

[15] Aaron M. Dollar and Robert Howe. A robust compliant grasper via shape deposition manufacturing. *IEEE/ASME Transactions on Mechatronics*, 11(2):154–161, 2006.

[16] Lael U. Odhner and Aaron M. Dollar. Fast, accurate models for predicting the compliance of elastic flexure-jointed robots. *Proceedings of the ASME Design Engineering Technical Conference*, 2, 01 2010.

[17] Lael U. Odhner and Aaron M. Dollar. The smooth curvature model: An efficient representation of euler–bernoulli flexures as robot joints. *IEEE Transactions on Robotics*, 28(4):761–772, 2012.

[18] Raymond R. Ma and Aaron M. Dollar. An underactuated hand for efficient finger-gaiting-based dexterous manipulation. In *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, pages 2214–2219, 2014.

[19] 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.

[20] C. Rosales, J. M. Porta, and L. Ros. Grasp optimization under specific contact constraints. *IEEE Transactions on Robotics*, 29(3):746–757, 2013.

[21] Antonio Bicchi and Vijay Kumar. Robotic grasping and manipulation. In Salvatore Nicosia, Bruno Siciliano, Antonio Bicchi, and Paolo Valigi, editors, *Ramsete. Lecture Notes in Control and Information Sciences*, volume 270. Springer, 2001.

[22] Elon Rimon and Joel Burdick. *The Mechanics of Robot Grasping*. Cambridge University Press, 2019.

[23] Amir Pagoli, Frédéric Chapelle, Juan Antonio Corrales, Youcef Mezouar, and Yuri Lapusta. A soft robotic gripper with an active palm and reconfigurable fingers for fully dexterous in-hand manipulation *. *IEEE Robotics and Automation Letters*, 6(4):7706–7713, 2021.

[24] Shenli Yuan, Austin D. Epps, Jerome B. Nowak, and J. Kenneth Salisbury. Design of a roller-based dexterous hand for object grasping and within-hand manipulation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8870–8876, 2020.

[25] Tong Cui, Jing Xiao, and Aiguo Song. Simulation of grasping deformable objects with a virtual human hand. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3965–3970, 2008.

[26] Yan-Bin Jia, Feng Guo, and Huan Lin. Grasping deformable planar objects: Squeeze, stick/slip analysis, and energy-based optimalities. *The International Journal of Robotics Research*, 33(6):866–897, 2014.

[27] Lazher Zaidi, Juan Antonio Corrales, Belhassen Chedli Bouzgarrou, Youcef Mezouar, and Laurent Sabourin. Model-based strategy for grasping 3d deformable objects using a multi-fingered robotic hand. *Robotics and Autonomous Systems*, 95:196–206, 2017.

[28] Tom Tsabar, Efi Fogel, and Dan Halperin. Optimized synthesis of snapping fixtures. In Steven M. LaValle, Ming Lin, Timo Ojala, Dylan Shell, and Jingjin Yu, editors, *Algorithmic Foundations of Robotics XIV*, pages 143–158, Cham, 2021. Springer International Publishing.

[29] Prosenjit Bose, Dan Halperin, and Shahar Shamai. On the separation of a polyhedron from its single-part mold. In *Proc. 13th IEEE Int. Conference on Autom. Sci. and Eng.*, pages 61–66. IEEE Comput. Society Press, 2017.

[30] Andreas F. Holmsen and Rephael Wenger. Helly-type theorems and geometric transversals. In Jacob E. Goodman, Joseph O'Rourke, and Csaba D. Toth, editors, *Handb. Disc. Comput. Geom.*, chapter 4. Chapman & Hall/CRC, 3rd edition, 2018.

[31] Jörg M. Wills. On polyhedra with transitivity properties. *Discrete & Computational Geometry*, 1:195–199, 1986.

[32] André Bouchet. Orientable and nonorientable genus of the complete bipartite graph. *Journal of Combinatorial Theory, Series B*, 24(1):24–33, 1978.

[33] Sébastien Loriot, Mael Rouxel-Labbé, Jane Tournois, and Ilker O. Yaz. Polygon Mesh Processing. In CGAL *User and Reference Manual*. CGAL Editorial Board, 5.0.1 edition, 2020.

תקציר

התיזה עוסקת בבעיית קיבוע (fixturing) של פאונים על ידי תופסנים ייעודיים. קיבוע היא משימה אופיינית בתחום הרובוטיקה והייצור. משתמשים בשיטות קיבוע כדי לחבר חיישנים לרובוטים, לבצע פעולות ייצור כמו כרסום או הרכבה ועוד. בעבודה זו אנחנו מתמקדים באפיון סוג חדש של תופסנים להם אנחנו קוראים בשם *תופסנים נעלים* (להלן snapping fixtures), בהוכחת תכונות על אותם תופסנים ופיתוח ומימוש אלגוריתם אשר מייצר מודלים דיגיטליים של התופסנים שניתן להזין ישירות למדפסות תלת ממד המדפיסות בחומרים פלסטיים. ל-snapping fixtures יתרונות ייחודים על פני שיטות קיבוע קלאסיות, כמו הסתמכות על תכונת הגמישות המאפשרת לכל תופסן להיות בנוי מחלק בודד ובכל זאת לאפשר הכנסת והוצאת הפאון אל הקיבוע.

המחקר מתעסק בשאלה הבאה:
בהינתן פאון P עם n קודקודים וגנוס (genus) בגודל חסום על ידי קבוע, האם קיים snapping fixture חצי-קשיח G העשוי מ-*כף יד* ו-*אצבעות* כך שניתן לדחוף את P לכיוון G ותוך כיפוף קטן של אצבעות התופסן G להגיע לתצורה חדשה שבה P ו-G בלתי ניתנים להפרדה כחלקים קשיחים (מקובעים).

המחקר מגדיר קריטריון אופטימיזציה ראשון של הבאה למינימום של כמות האצבעות בתופסן ומאפשר קריטריוני אופטימיזציה נוספים. אנו מוכיחים כי תמיד ניתן לייצר snapping fixture בעל לכל היותר 4 *אצבעות* עבור פאון שניתן לקבע בשיטה זאת, אך מביאים דוגמאות נגדיות של פאונים עבורם לא ניתן לייצר snapping fixtures כלל.

אנו מציגים אלגוריתם בסיבוכיות זמן ריצה של $O(n^3)$ אשר מייצר snapping fixture עבור פאון קלט P בעל n קודקודים, מביא למינימום את כמות האצבעות, תומך בפונקציית אופטימיזציה נוספת, ומחזיר תשובה שלילית אם לא קיים snapping fixture עבור P. בנוסף מימשנו אלגוריתם פשוט יותר שרץ בסיבוכיות $O(n^4)$ אשר מייצר מודל תלת מימדי של התופסן ומנגישים אותו לציבור הרחב בעזרת אתר אינטרנט ייעודי שהקמנו.

לבסוף אנו מציגים פרטי מימוש, שימושים, תוצאות ניסויים והנחיות להדפסת תופסנים במדפסות תלת מימד.

מאמר המסכם את תוצאות התזה התקבל לפרסום והוצג בכנס
14th International Workshop on the Algorithmic Foundations of Robotics (WAFR), 2020.

ייצור אופטימלי של תופסנים ננעלים

(snapping fixtures)

חיבור זה מוגש כחלק ממילוי הדרישות לקבלת
התואר "מוסמך למדעים" (M.Sc.) בבית הספר למדעי המחשב
באוניברסיטת תל-אביב
ע"י

תום צבר

העבודה הוכנה באוניברסיטת תל-אביב
בהדרכת פרופסור דן הלפרין
[אדר] התשפ"ב