Geometric Algorithms for Part Orienting and Probing

Fatemeh Panahi

.

This thesis was typeset by the author in LaTeX 2ε. The main body of the text was set using 10 points Palatino font.

ISBN: 978-94-6299-253-5

Geometric Algorithms for Part Orienting and Probing

Geometrische Algoritmen voor het Oriënteren en Aftasten van Onderdelen

(met een samenvatting in het Nederlands)

Proefschrift

ter verkrijging van de graad van doctor aan de Universiteit Utrecht op gezag van de rector magnificus, prof. dr. G.J. van der Zwaan, ingevolge het besluit van het college voor promoties in het openbaar te verdedigen op dinsdag 22 december 2015 des middags te 12.45 uur

door

Fatemeh Panahi

geboren op 19 februari 1985
te Zanjan, Iran

Promotor: Prof. dr. M. van Kreveld
Copromotor: Dr. ir. A. F. van der Stappen

This work was supported by the
Netherlands Organisation for Scientific Research NWO,
grant number 612.001.016 (AMPLIFI).

Contents

PART I

INTRODUCTION

Chapter 1

Introduction

"It is the glory of geometry that from so few principles, fetched from without, it is able to accomplish so much."

- Isaac Newton

In the field of algorithmic automation, there are a number of different tasks, many of which can benefit from techniques from computational geometry. In order to design processes for these tasks, one has to precisely formulate them in mathematical terms while taking their physics and environmental restrictions into account. Although each of these problems is formalized in a different way, they all have in common that they deal with parts. The shape of these parts determines to a large extent how automation tasks are accomplished. As a result, geometry plays a major role in the design of plans for the tasks. Computational geometry is about dealing with geometric problems and designing provably efficient algorithms, so it can help to solve automation problems efficiently.

In the assembly process, one often needs to move parts from one orientation to another. There are two basic approaches for ensuring that a part is in some particular position and orientation. One is to use a vision system to sense this information. Another approach is to use mechanical means. The purpose is to act on the part in a manner that reduces uncertainty. Another important task in manipulation and robotics is the process of learning the shape and orientation of an unknown object using a simple measurement device called *probe*. Manipulation tasks such as part orienting generally take

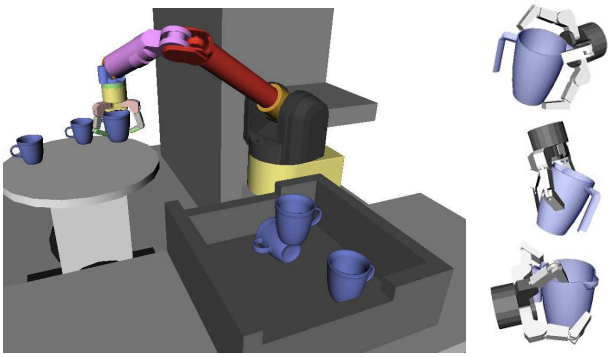

Figure 1.1: The task is to grasp the cups in the same shape but arbitrary orientations and put them on the table. (Digital image, OpenRAVE, web, <openrave.org>.)

place in structured environments where we cannot perform intricate measurements on the parts. The RISC (Reduced Intricacy in Sensing and Control) approach is an attempt to design cost effective and reliable systems which are easy to set up and reconfigure [48].

Rather than using complex manipulators and sensors, RISC systems are designed based on simple hardware components which are capable of performing only simple physical actions or using sensors with low dimensional output. In these systems, automation hardware such as parallel jaw grippers might be used instead of multi fingered hands and optical beam sensors rather than cameras.

The geometric shape of an object and its orientation are major parameters in the design solutions for many automation tasks. See Figure 1.1. At the intersection of automation and computational geometry, and in light of the RISC approach, we have determined a collection of research problems with immediate applications in automation (particularly in manufacturing and metrology). This thesis has two main parts: the first part investigates different problems related to orienting parts; in all these problems we assume that the geometric shape of the parts is known; e.g. it is given as an input. The second part discusses a novel type of probing which is used to measure the shape of an unknown object.

1.1 Orienting parts

Robotic manipulation in industrial automation often deals with various part handling problems. One such problem, which arises in automated assembly,

is the so-called part feeding or orienting problem. Manufacturing processes mostly produce a stream of parts that must be reoriented before assembly on industrial production lines. A part feeder is a machine which takes a stream of identical parts with arbitrary initial orientation or *pose* and orients them to a unique final pose. (See Figure 1.2.)

Part feeders may rely solely on the geometry of specially designed fixtures interacting with a part on a conveyor belt or in a gravity field, or specially designed motions of generic surfaces or some combination of geometry, materials, and motion (open loop or sensor based) design [100]. In all cases, the goal is to collapse the possible initial orientation of the part into a smaller set (ideally a singleton).

Sensorless manipulation has received considerable attention over the past two decades. More specifically, part feeders have been studied in the line of thought of the RISC approach in which no sensory information, or low dimensional output sensing are used to orient the part [15]. The shape of the part is given as an input and and the output is a sequence of actions that moves the part from an unknown initial pose into a unique final pose.

Two classes of orienting devices exist. There are devices which reject parts which are incorrectly oriented; also there are devices which physically reorient the parts in the desired manner. Current technology in part feeders relies heavily on rejection techniques [130]. These techniques randomize the orientation of the parts, mostly by shaking or dropping them, and then reject all those parts which are not in the correct orientation. The rejected parts are recirculated, and this process repeats indefinitely. Research into the probability distribution of stable part orientations suggests that for a typical part, a minority of the parts will randomly assume the correct orientation. Therefore, rejection-based methods are very inefficient [43]. In this subsection, we overview some of the most important sensorless approaches for orienting parts and then discuss the distribution of final orientations of a part being dropped on a surface with random initial orientation.

Figure 1.2: A part feeder takes parts in initially arbitrary orientation and orients them in a unique final orientation.

1.1.1 Different approaches for part orienting

There is a rich literature on object orientation and a variety of sensorless part feeders have been proposed. The first algorithm for orienting parts without sensors was proposed by Erdmann and Mason [62]. Their approach was by controlling the slip motion of a planar tray in a specific sequence of directions, causing a planar part to slide along walls and into corners in order to reduce the number of possible orientations. They described critical tilting directions and developed a complete algorithm to find tilt sequences. Later, Natarjan [109] and also Erdmann et al. [63] focused on three-dimensional polyhedral objects resting on a tiltable table and proposed a planner that determines a sequence of wobbling operations designed to minimize the uncertainty in the part's orientation. Swanson et al. [130] proposed a flat vibratory table as a means of orienting parts. They sought to design a table motion which should work by bouncing parts gently on the table vertically while inducing momentary horizontal forces at the contact points which cause a torque to be applied to the center of mass.

Mason [101] was the first to identify the role of *pushing* and *sliding* as fundamental to manipulation. An elegant result of his analysis is that one can determine a simple rule for the rotation of an object in contact with a point pusher, using only the location of the object center of mass, the push direction, the contact point and normal, and the coefficient of friction between the pusher and pushed object.

One of the most fundamental approaches to part orienting, which is also partly our focus in this thesis, is through a sequence of push actions by means of a single jaw [4, 99, 118]. Each push action is in a direction orthogonal to the length of the jaw, and the orientation of the jaw between pushes is independent of the orientation of the part.

Squeeze actions by a parallel jaw gripper have also been applied for orienting a part. An action consists of orienting the gripper, closing the jaws as far as possible when the object lies between them, and finally opening the jaws again. Goldberg [70] showed that there always exists a plan (a sequence of pushing and squeezing actions) for orienting a polygonal part using a frictionless single jaw or parallel jaw gripper and proposed a greedy algorithm for computing the shortest such plan in $O(n^2 \log n)$ time where n is the number of vertices of the part. He was the first to present bounds on the number of actions. His algorithm requires $O(n^2)$ actions in the worst case. He conjectured that the length of the shortest plan is linear in n. Chen and Ierardi [50] proved Goldberg's conjecture and also showed how to compute the maximum radius of a disk inside which the vertices of a polygonal part can vary such that a plan still exists. In addition, Goldberg [70] and Chen and Ierardi [50] studied grasps in which one jaw first pushes the part and then both jaws squeeze it. Van der Stappen et al. [133] showed that even though

orienting parts require $O(n)$ push actions, they deduce a new bound on the length of the shortest push plan that depends on the thinness or eccentricity of the part. The bound shows that only $O(1)$ pushes are required for the large class of parts with non-square minimum width bounding boxes. Berretty et al. [16] showed that 3D (polyhedral) parts can be oriented by a sequence of pushes by a perpendicular pair of planar jaws and gave an $O(n^3 \log n)$ time algorithm to find such a plan.

There are also approaches that are based on constrained forms of pushing. A sequence of fences can be attached to a conveyor belt and parts slide along these fences which are designed to passively reorient parts as they travel down the belt. The motion of the belt effectively turns each slide into a push action by the fence in the direction normal to the fence with the effect of aligning the parts. Note that it is because the system is assumed to be frictionless. The problem of designing the fences is equivalent to computing push actions with constraints on successive push directions. If the belt moves downward, then any fence encountered by a part will effectively perform a push with positive vertical component; thus, only half of the possible push directions are available at a given time. It makes the problem more complicated, and requires a different approach.

The *fence design* problem was first considered by Peshkin and Sanderson [117]. Wiegley et al. [139] added curved tips to the fences to control the way in which the part leaves the fence. They gave a complete algorithm to compute the shortest sequence of fences. Their algorithm guarantees to find a design if one exists and to terminate with a negative report otherwise. The running time is exponential in the complexity of the part to be oriented. Also, they conjectured that a polynomial time fence design algorithm exists for any polygonal part. Berretty et al. [22], gave an $O(n^3 \log n)$ time algorithm to find such a sequence for any polygonal part with n vertices. Their time and complexity bounds are similar to those for pure pushing. Gudmundsson and Goldberg [73] used a queuing model to investigate optimal conveyor belt velocities.

Parts on a *conveyor belt* can be oriented also by a single moving fence instead of a sequence of stationary fences. Akella et al. [3] examined a one joint robot which sweeps a planar fence over the conveyor belt.

A sequence of fixed horizontal pins suspended above the belt at different heights can be used to topple a three-dimensional part as it moves on a conveyor belts. Zhan et al. presented an algorithm to design a sequence of pin locations that will guarantee that the part will enter a unique final orientation or to determine that no such sequence exists [141].

Parts can also be oriented by sequence of *pulling* actions. Berretty et al. [18] investigated the problem of polygonal part feeding by pulling fingers. A pulling finger is assumed to be a point contact inside a polygonal

part pulling it outwards. They proposed an algorithm to find the shortest sequence of pull actions for any asymmetric convex polygonal part that will put it into a single final orientation.

Zumel and Erdmann showed that parts can also be oriented by *sliding* and *rolling* between two low friction non-prehensile palms [142, 143]. The palms are joined at a central hinge and support a part in a gravitational field. The system computes and executes a sequence of palm motions designed to reorient the part.

An alternative technique to part orienting is to use *throwing and catching (juggling)*. Lynch et al. [100] described a system with a one-joint planar arm combined with a simple sensor which can be used to recognize when the part has reached the goal orientation. This system can be used as a part feeder by repeatedly throwing and catching parts on a surface. They showed that for any planar parts, a proper choice of throw velocity and arm geometry, will bring the part to the goal configuration.

Blind et al. [34] investigated an approach which drops polygonal parts on a simple device consisting of a grid of retractable pins mounted on a vertical plate. It is possible that a part dropped on this device comes to rest on the actuated pins, bounces out, or falls through. The goal is to find a sequence of pin actuations that will cause parts to emerge into a unique orientation after a number of dropping actions.

Among the various sensorless part feeders used in industry, the most versatile and still widely used method is the *vibratory bowl feeder*. One of the most comprehensive works on the design of parts feeding and assembly design is a book by Boothroyd et al. [38], which describes vibratory bowls as well as non-vibratory parts feeders in detail. Parts in a bowl are gently shaken using a rotary motion so that they climb a helical track. Traditional bowl feeders rely heavily on rejection techniques. These techniques randomize the orientation of the parts, often by shaking or dropping them, and then reject all those parts which are not in the desired orientation. The rejected parts are forced back to the bottom of the bowl to recirculate, and the process repeats and parts will gradually be shaken so that finally they are all aligned, caused by sequence of mechanical filter devices such as grooves, traps (gaps and narrowed tracks), pins and wiper blades, in the track that are shaped to fit the part [37, 97]. There is a wide range of various approaches for the design of bowl feeder layouts. We will focus on some recent geometric approaches.

Traps are a class of vibratory bowl filters that are obtained by removing sections from the track [1, 19–21, 132]. Berretty et al. [19–21], proposed several classes of traps such as balconies, gaps, slots, canyons, and general polygonal traps and gave algorithms for designing those traps. Agarwal et al. [1] described the design of minimal traps to filter polygonal parts. Goemans et al. [67, 69], introduced additional traps with specific guarantees (guillotine

traps for polygonal parts and V-shaped traps for polyhedral parts) and also proposed efficient algorithms that report the complete set of valid traps.

Another class of geometric filters of interest for vibratory tracks are *blades*. A simple primitive, consisting of one horizontally mounted convex polygonal metal blade, that can feed a broad class of three-dimensional parts has been developed in [66, 68]. The proposed complete procedure identifies all single blade solutions that feed any given 3D part.

Programmable force fields offer a different approach for a class of devices for distributed, non-prehensile manipulation. A force field is a vector field indicating the forces exerted by one object on another. Instead of handling parts directly, one can be move them by inducing a force field on the surface that supports the part. In a programmable force field, the forces generated at each point of the field can be controlled independently. For a detailed review see [29, 30, 88]. Unlike robot grippers, conveyor belts, or vibratory bowl feeders, these devices generate force vector fields in which the parts move until they may reach a stable orientation. Several devices have been introduced that can implement programmable force fields. An example of those is actuator arrays which are able to create vector force fields using Micro Electro Mechanical System (MEMS) technology [28]. Another example is a vibrating plate system [26] which consists of an aluminum plate that is attached to an electrodynamic vibration generator. The basic commonly used vector fields are described in [64]. The uniform field moves an object linearly, the squeeze field aligns it to a common edge of the surface elements, the radial field aligns it to a common corner of several surface elements and the tangential field rotates an object. Squeeze fields have been studied extensively as a strategy to orient parts [23, 25, 31, 84]. Combinations of squeeze fields and radial fields have been used to reduce the length of the sequence of vector fields required for orienting the part [24, 27].

1.1.2 Pose statistics

Pose statistics is the analysis of the complex behavior of objects being dropped on a surface. When an industrial part is dropped on a surface, it settles into an orientation such that it does not topple over under the influence of gravity; this is called a *stable pose*. Note that in the 2D case, we assume that planar (slab-like) objects roll along their circumference. Pose statistics is the study of identifying the stable poses of an object and estimating the likelihood that the object will arrive in each of these poses, i.e. finding the distribution of the final poses. Results from studying pose statistics are substantial for the design of many automated tasks specially for the design of part feeders. Not only a sequence of dropping actions is considered as an approach to orient a part, but also we note that parts are usually dropped on a work-surface

which can be a tray, a conveyor belt or a vibratory bowl feeder. Therefore, it is useful to have an understanding of the ways and probabilities in which parts will arrive on the supporting surface. For example, assume a particular part being dropped on a surface has two possible stable poses with high probabilities. One pose feeds the part correctly while the other causes assembly failure. Pose statistics approach computes these two poses and the probabilties corresponding to these poses. When the approach is for example using vibratory bowl feeder it can result in a faster and more effective design of part orienting.

There are a number of works estimating the probability distribution over a part's stable poses where its initial orientation is uniformly at random and the part falls on a flat work-surface in presence of gravity [71, 103, 140]. Both theoretical and experimental methods are used for determining the likely stable state of parts with different shapes. Boothroyd et al. [39] have proposed a method for rectangular prisms. The energy barrier method, proposed by Boothroyd and Ho [40] considers the amount of kinetic energy a part may have in a certain pose and also the amount of kinetic energy it needs to leave that pose. The probability that a part will attain a particular stable pose was determined to be a function of these two factors. This method was developed for a selected class of parts consisting of both regular prismatic and cylindrical geometries.

Extending Boothroyd's work, Ngoi and Lim [112] presented the energy envelope method which integrates the work done for moving a part in a stable pose to all possible landing positions. The probability that a part comes to rest in a particular stable pose is proportional to the energy envelope of that pose. The centroid solid angle method and displacement center of gravity method are alternative methods proposed by Ngoi et al. [110, 111]. These methods are based on the hypothesis that the probabilty of a component resting on a specific pose is directly proportional to the magnitude of the centroid solid angle and inversely proportional to the height of its centroid from that pose. Chua and Tay [51] proposed a mathematical model for predicting the final poses for small parts with regular shapes. Their approach is based on stability considerations.

There are also analyses on more general classes of parts. Wiegley et al. [140] proposed an algorithm for estimating the distribution of stable poses of a polyhedron under quasi-static conditions (i.e. without dynamics). Mirtich et al. [71, 103] developed this algorithm further by considering a simple model of dynamic stability. In addition to a result from a quasi-static approach, they discussed the results from full dynamic Monte Carlo simulation and they compared the algorithms to real data. Considering polyhedral parts as well as curved parts, Kriegman [87] defined the notion of a *capture region*, the region in configuration space in which any initial configuration will converge

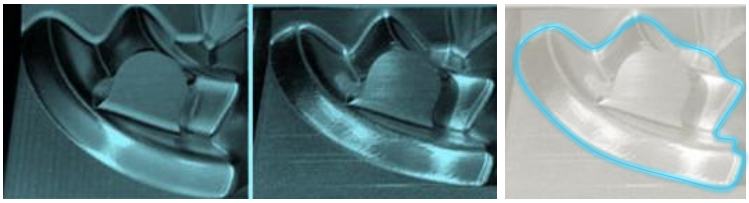

Figure 1.3: An example of two industrial parts with similar but not identical geometric shape. (Digital image, International Haas Automation Inc, web, <haascnc.com>.)

to a given final configuration. Moll and Erdmann [105, 106] studied the pose distribution for an object being dropped on a sloped surface or into a curved bowl. In particular, they showed that by controlling the height and initial velocity of the part, and the shape of the bowl, it is possible to greatly reduce uncertainty on the final orientation of the part. Suresh et al. [128] investigated the most probable final pose of a part by dropping it from different heights. The favorable orientation obtained through drop testing has been compared with the likely orientations identified through theoretical methods identified in the literature. Recently, Várkonyi [134] created a large simulated data-set and evaluated the estimators proposed in the literature by comparing their predictions to simulation results; he also proposed new estimation algorithms. If an object has just one stable pose it is called *monostatic*. Arnold [7] conjectured that there exist 3D homogeneous, convex monostatic objects. Várkonyi and Domokos [135] proved this conjucture and called the object *Gömböc*. However, in the 2D case, it has been shown that no monostatic homogeneous planar objects exist [58]. This statement is equivalent to the well-known Four-Vertex Theorem [12] in differential geometry.

1.2 Parts with shape variation

Most of the existing solutions in algorithmic automation assume a severely idealized world in which parts are perfectly identical to their CAD model, and manipulators and sensors are infinitely accurate. In real life, however, industrial parts are manufactured to tolerances [121, 137] as no production process is capable of delivering perfectly identical parts. Variations in part shape [45], inaccurate sensors [90] and actuators [57] cause the aforementioned algorithms to fail when employed in practice. The challenge is therefore to design algorithms for planning manipulation tasks that explicitly take into account manipulator (and sensor) inaccuracy and part imperfection, and report solutions that work despite their presence. In this thesis we particu-

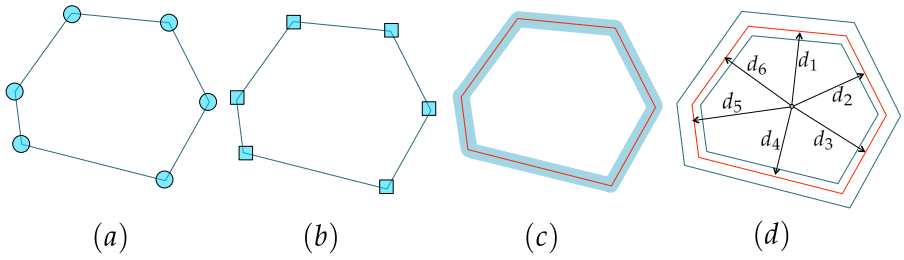

Figure 1.4: Different geometric models for shape variation: vertices can vary inside a convex region such as a (a) disk or (b) square, (c) edge variation is bounded by their distance from the origin, It shows a particular instance of the part, the variational parameters $d_1, ..., d_6$ and the extreme positions of the edgesupporting lines. (d) solid offset: the boundary of object can vary within a given distance.

larly focus on shape variation of a part. Figure 1.3 illustrates an example of shape variation for two instances of an industrial part. Note that variations in shape and center of mass are not the only sources of uncertainty in robotics. Additional uncertainty can result from the inaccuracy of the actuators and manipulators [57] and sensors [90].

In order to conduct shape variation analysis, an appropriate model has to be defined. In the current industrial practice, standards [9, 10] define dimensioning practices and provide mathematical tolerance zone definitions for geometric features. Due to its complex nature, the standard has not indicated yet how to model the variations.

Several geometric approaches and representations have been considered to model variation in geometric data. The simplest way is to bound the point variations by a region. The model of ϵ-geometry assumes that a point can vary within a disk of radius ϵ [5, 75, 85] or a rectangle [49]. Region-based models represent a point by any convex region [44, 54, 95]. Edge variations can be bounded by their distance from the origin [78, 89]. Another approach is offsetting the boundary of an object which is called solid offset. To solid offset an object P by a positive distance ϵ one adds to the nominal object all the points exterior to P that lie within a distance ϵ of the boundary of P. For a negative solid offset, all the points of P within a distance ϵ from its boundary are subtracted from the nominal object. See Fig 1.4.

In the aforementioned geometric models the only information that we use is the region in which the points lie. These models are too restrictive if there is more information available. For instance, in other approaches dependency of the points' position on a number of parameters [76, 81, 123] or probability distribution of the location of the point [96] have been considered.

There have been a few studies into part feeding in the context of parts with shape variation. Generally, in all the models which have been considered each vertex is represented by a region in which it may vary. Akella et al. [5] studied the problem of orienting convex polygons whose vertices and center of mass lie inside a disk. They required that any variation keeps the part convex. They proposed graph-based approaches for fence and push-squeeze plans for parts that satisfy their assumptions. The problem of orienting a part by fences by considering shape variation has been studied by Chen et al. [46, 49]. They used a similar model for part shape variation by allowing the vertices to vary inside disks and squares that are defined relative to the center of mass. They proposed a method for computing the maximum allowable disk or square for variation of each vertex such that the existing plan still works.

Variation in the shape of a part also leads to variation in the location of its center of mass. Due to the key role of the center of mass in many applications such as pushing, grasping, or pulling, bounding the set of possible locations of the center of mass for an imperfect part is an important problem. Bern et al [13] studied the problem of centroid for a set of points with approximation weights. Planning algorithms should take into account variation in shape as well as the location of the center of mass to prevent failure when the resulting plans are applied to manufactured incarnations of a model part. So, it is important to study the relation between variation in part shape and variation in the location of the center of mass for a part with uniform mass distribution.

Akella et al. [5] estimated the locus for a polygon under the ϵ-geometry model. The problem of finding the locus of the center of mass of a part with shape variation and uniformly distributed mass has been mentioned as an open problem [5, 46].

In comparison with the aforementioned studies, we consider a more general model for shape variation that allows to characterize variation along the entire boundary instead of only at the vertices. The model assumes that any valid instance of a part contains a given closed shape while it is contained in another given closed shape. Our goal is to orient a part with shape variation, that is, we want to find the sequence of pushes that puts all instances from the shape family into the smallest possible interval of orientations. We also study the problem of finding the locus of the center of mass for all instances of a part satisfying our model for shape variations.

1.3 Probing objects for metrology

Metrology, the study of measurement, has applications in manufacturing, inspection, robotics, surveying, and healthcare [53, 60]. An important aspect of metrology is the problem of how to most efficiently use a given measurement

device to obtain a specific piece of complex information. There are different approaches to measuring shape of an object such as virtual metrology and statistical methods, high-dimensional output sensing (such as a camera). If we use only low-dimensional output sensing which reports a low-dimensional measurement (such as a distance or coordinate) then the approach is called *probing*. An example of a very well-known device for measuring the physical geometrical characteristics of an object is the *Coordinate Measuring Machine (CMM)*. Measurements are defined by a probe attached to the third moving axis of this machine.

When the measurement device and object of interest are geometric, the problem of obtaining information about the object through repeated use of the device is known as *geometric probing*. A common version of this problem is to deduce the shape of an unknown object using as few probes as possible.

Efficient algorithms for probing convex polytopes have been the subject of many papers, particularly the task of minimizing the complexity (in terms of number of probes required for measuring the shape of the polytope). Geometric probing describes many mathematical and practical problems, and has found applications in robotics, computational tomography, and medical instrumentation [125, 126]. Thanks to its wide range of applications, it has been extensively studied ever since it was first formally posed by Cole and Yap [52]. The problem in general is interactive, where the probes are determined in sequence with each probe choice guided by the results of the previous probes. In a model-based testing problem, a finite set of objects is given, and the problem is to find a number of probes to recognize which model from the given set is presented.

Within geometric probing, a wide range of probes, geometric objects, and objectives have been studied. In particular, the problem of probing convex polytopes has been the subject of a wealth of research, starting with Dobkin et al. [56], who studied the complexity (in terms of number of probes required) of reconstructing a convex polytope for many different types of probes. These include a *point probe* [79] which determines whether a given input point is inside or on the boundary of the object or not. A *finger probe* models a robotic finger moving along a given input ray and measures the intersection point with the boundary of the object to be probed. It has been shown by Cole and Yap [52] that a convex polygon with n vertices can be deduced using no more than $3n$ finger probes and $3n - 1$ probes are sometimes necessary. Boissonnat and Yvinec [33] assumed a more general probe model which also returns the normal at the contact point for non-convex polygons. They proved that $3n - 3$ probes are sufficient under the assumption that there are no collinear edges. For model-based testing, it has been shown that $2n + 1$ finger probes are sufficient to identify a convex polygon [14]. This result has been improved to $n + 4$ and it was shown that $n - 1$ probes are necessary [80]. Guha

and Khanh [74] proposed an approach which recognizes an object with high probability from a given model set which uses a constant number of finger probes.

Line probes (or *hyperplane probes* in higher dimensions) are another type of probes which slide a line (or plane) in a particular direction until it intersects the object. A variation on line probes is a half-line that pivots around a given input point until it hits the object. A *cone probe* is the combination of two half-line probes pivoting in opposite directions until they touch the object on both sides. A *projection probes* is a mathematical projection of a planar object onto a line; Li [92] showed that a convex polygon can be determined with $3n$ line probes and $3n - 2$ projection probes. Skiena [126] observed that there is duality between finger and line probes, therefore there is a one-to-one correspondence between algorithms that use finger probes and ones that use line probes.

Rao and Goldberg [120] and Arkin et al. [6] studied *diameter probes* which measure diameter using a parallel jaw gripper. They proved that it is possible to recognize a shape in the case of model-based testing i.e. recognizing from a set of known objects.

Another type of probe which is called *X-ray probe* gives the length of intersection between a given line and the object. It was shown that $5n + O(1)$ probes are sufficient for determining a convex polygon and $(3n - 1)/2$ are necessary. Meijer and Skiena [102] also studied *histogram probes* which return a histogram representing the length of intersection between the object and a line at a given angle as the line is swept through the entire object.

Furthermore, *half-plane probes* through an object compute the area of intersection between a half-plane and the object. Skiena [127] developed techniques based on x-ray probing to determine convex n-gons using $7n + 7$ half-plane probes. He showed that there is a linear lower bound for determining a convex polygon.

There are also studies which have looked at the problem of approximating non-polygonal convex objects with respect to some metric [94, 122]. Lindenbaum and Bruckstein [94] raised this problem and in particular, they considered the problem of approximating an arbitrary bounded convex set from hyper-plane probes, with respect to the Hausdorff metric (Hausdorff distance is the greatest of all the distances from a point in one set to the closest point in the other set).

Besides the application of probing in reconstructing the shape of an object, probing has been extensively studied as a tool to learning smooth surfaces and also determining the physical properties at the nanoscale (see for example [36, 82]). During microscale and nanoscale fabrication, it is similarly necessary to determine the shape of planar structures. To this end, a variety of techniques have been developed. Scanning probe microscopy (SPM) is a

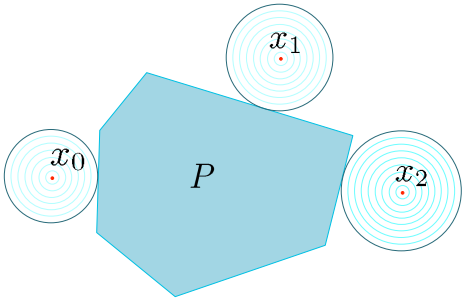

Figure 1.5: An unknown polygon P with proximity probes at x_0, x_1, x_2

branch of microscopy that forms images of surfaces using a physical probe that scans the specimen. A probe moves relative to the sample surface and measures the interaction.

In this thesis, we introduce and study a novel type of probe called *proximity probe*, for convex polygons. The proximity probe returns the distance from a chosen point to the nearest point of the unknown polygon, provided the chosen point does not lie in the interior of the polygon (if it does, it returns *error*). This probe can be viewed as modeling a device which expands a disk around the chosen point until it collides with the polygon and then measures the final radius of the disk. For simplicity, and because it is not a very restrictive assumption, we will assume that the unknown polygon lies inside a known bounded region. Figure 1.5 shows the results of three distance probes x_0, x_1, x_2 measuring the unknown polygon P. We then present efficient algorithms using this probe both for the problem of reconstructing P and the related problem of identifying P from a finite model set.

Our original inspiration for studying proximity probes is the basic concept of sonar detection, in which the sensor emits a ping and listens for the echo off of nearby objects. In the simplest case, the sensor merely registers the first time it hears an echo, which can be converted into the distance (though not the direction) between the sensor and the nearest object. Thus, this problem has potential applications in robotics, especially in cases where severe power, size, weight, or cost constraints might require that a low-dimensional output sensor be used rather than more sophisticated alternatives.

1.4 Contributions and outline of the thesis

In this thesis, detailed solutions are presented to several problems dealing with geometric shape and orientation of an object in the field of robotics and

automation. The solutions fit in the RISC framework in the sense that they rely on simple actuators and sensors.

The remainder of the thesis consists of two main parts. In Part I, the problems related to orienting an industrial part are considered. In Chapter 2, in comparison with the aforementioned studies we define a more general model for shape variation that allows to characterize variation along the entire boundary of the object instead of only at the vertices. This model for shape variation assumes that every valid instance contains a shape P_I while it is contained in another shape P_E such that $P_I \subseteq P_E$. The contained and containing shapes we use bear resemblance to Requicha's Least Material Condition (LMC) and Maximum Material Condition (MMC), respectively, which were defined by means of offset solids [121].

We study the problem of orienting a part with given admitted shape variations by means of pushing with a single frictionless jaw when P_I and P_E are convex polygons. The problem that we solve is to determine, for a given h, the sequence of h push actions that puts all valid instances of a part with given shape variation into the smallest possible interval of final orientations. The resulting algorithm runs in $O(hn)$ time, where $n = |P_I| + |P_E|$.

Chapter 2 is based on the following publications.

- *F. Panahi, M. Davoodi, A. F. van der Stappen, Orienting parts with shape variation, Proc. of the 11th Workshop on the Algorithmic Foundations of Robotics, pp. 479-496, 2014.*
- *F. Panahi, M. Davoodi, A. F. van der Stappen, Orienting parts with shape variation, IEEE Transactions on Automation Science and Engineering, 2015, to appear.*

In Chapter 3, we consider the same model for shape variation and we study the relation between variation in part shape and variation in the location of the center of mass for a part with uniform mass distribution. We characterize the worst-case displacement of the center of mass in a given direction in terms of P_I and P_E. The corresponding displacement can be used to obtain a k-facet outer approximation of the set of all possible loci of the center of mass in $O(kn \log n)$ steps where n is the complexity of P_I and P_E. It is shown that the worst-case displacement is small if P_I is convex and fat (that is, not long and thin) and the distance between the boundaries of P_E and P_I is bounded. It presents a general approach to handle a broad class of both two- and three-dimensional semi-algebraic parts.

Chapter 3 is based on the following publications.

- *F. Panahi, A. F. van der Stappen, Bounding the locus of the center of mass for a part with shape variation, Proc. of the Canadian Conference on Computational Geometry, pp. 247-252, 2013.*

- *F. Panahi, A. F. van der Stappen, On the location of the center of mass for parts with shape variation, Proc. of the IEEE /RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4482-4488, 2014.*

- *F. Panahi, A. F. van der Stappen, Bounding the locus of the center of mass for a part with shape variation, Computational Geometry: Theory and Applications, pp. 847-855, 2014.*

- *F. Panahi, A. F. van der Stappen, Reprint of: Bounding the locus of the center of mass for a part with shape variation. Computational Geometry: Theory and Applications 48(5), pp. 398-406, 2015.*

In Chapter 4, we explore the pose statistics problem for a family of 3D objects with initial pose uniformly at random. We assume that the object falls onto a flat surface in presence of gravity under quasi-statics conditions. We propose a novel type of geometric eccentricity for d-dimensional objects and show that the final pose distribution for 3D eccentric object has a substantial bias towards a small subset of poses. This result can be applied to part feeding tasks to obtain a faster design for reducing the object pose uncertainty. According to our proposed notion of eccentricity, for a given k, we have two types of eccentricity in 3D: k-(1)-eccentric objects are k times bigger in one dimension than the other two dimensions while k-(2)-eccentric objects are k times larger in two dimensions than in the other dimension. We show that for both types of eccentric objects, there is a small cluster of poses at which the object always ends up. We show that there is a high probability of ending up at a pose which is close to a specific plane for k-(1)-eccentric objects, and close to a specific line for k-(2)-eccentric objects. In addition, we investigate that for larger k, there is smaller range of poses (as a function of k) at which the k-(1)-eccentric and k-(2)-eccentric objects end up with higher probability.

Chapter 4 is based on the following publication.

- *F. Panahi, A. Adler, A. F. van der Stappen, Pose Statistics for Eccentric Parts, Proc. of the IEEE Conference on Automation Science and Engineering, to appear, 2015.*

In Part III, we move away from orienting parts and consider problems related to probing an object to measure the shape and orientation of that when it is unknown. In Chapter 5, we consider a *proximity probe* which, given a point, returns the distance to the boundary of the nearest object. When there is an unknown convex polygon P in the plane, the goal is to minimize the number of probe measurements needed to exactly determine the shape and location of P. We present an algorithm with upper bound of $3.5n + k + 2$ probes, where n is the number of vertices and $k \leq 3$ is the number of acute angles of P. The algorithm requires constant time per probe, and hence $O(n)$

time to determine P. We also address the related problem where the unknown polygon is a member of a known finite set Γ and the goal is to efficiently determine which polygon is present. When m is the size of Γ and n' is the maximum number of vertices of any member of Γ, we present an algorithm with an upper bound of $2n + 2$ probes with $O(1)$ computations per probe and a $O(n'm)$ preprocessing phase (depending only on Γ).

Chapter 5 is based on the following publications.

- *F. Panahi, A. Adler, A. F. van der Stappen, K. Y. Goldberg, An efficient proximity probing algorithm for metrology, Proc. of the IEEE Conference on Automation Science and Engineering, pp. 342-349, 2013.* Finalist for Best Student Paper Award.
- *A. Adler, F. Panahi, A. F. van der Stappen, K. Y. Goldberg, Efficient proximity probing algorithms for metrology, IEEE Transactions on Automation Science and Engineering 12(1), pp. 84-95, 2015.*

PART II

ORIENTING PARTS

Chapter 2

Orienting parts with shape variation

"As far as the laws of mathematics refer to reality, they are not certain; and as far as they are certain, they do not refer to reality."

- Albert Einstein

MANUFACTURED parts always have slight imperfections; hence, they are designed up to certain tolerances. However, most of the existing solutions for manipulation tasks are designed based on the assumption that the shape of a part is exactly the same as its CAD model. It is unacceptable that a plan for a manipulation task that was determined on the basis of a CAD model of a part fails on some manufactured instance of that part, and therefore it is crucial that the admitted shape variations are systematically taken into account during the planning of the task.

In Chapter 1 we have seen that there are only a few works that consider shape variation for manufactured parts and all of them consider variations of the vertices only. In this chapter, we define a more general model for shape variation that allows variation along entire boundary of the part. We consider this model for shape variation and focus on the problem of orienting a part by a sequence of push actions. In the part orienting or feeding problem a stream of parts are given in arbitrary initial orientation and the goal is to find a sequence of actions that align the parts into a unique final orientation. As it mentioned in Chapter 1 it has been shown that for perfectly identical

pasts, there always exists a sequence of pushes which orient them into a specific orientation [70]. For the version of parts with shape variation we are interested in finding the smallest set of final orientations. Our goal is to solve the part feeding or orienting problem for the imperfect part, that is, we want to find the sequence of pushes that puts all instances from the shape family into the smallest possible interval of orientations. To this end we generalize the notions of radius and push function [70] to families of shapes.

The remainder of the chapter is organized as follows. In Section 2.1 we introduce our model for part shape variations and the relevant functions that describe the shape and behavior of such parts. In Section 2.2 we define types of orientations and their properties. In Section 2.3 we bound the final orientation of the part after a push action using the properties of the orientations. In Section 2.4, we develop a greedy algorithm for reporting the smallest interval of possible orientations for the entire shape family after a given number h of pushes. In Section 2.5 we show that there exist imperfect parts for which there always is a next push that shrinks the interval of possible orientations. Finally we conclude in Section 2.6.

2.1 Preliminaries

In this section, we explain our assumptions and introduce the terminology and notations used throughout this chapter. We first describe the problem that we are going to study. Then, we will have a short review of the relevant concepts from previous work and finally we define similar concepts for a part with shape variation.

2.1.1 Shape variation model

We use a very general model for permitted shape variations that only requires that any manufactured instance of a part P_M contains a given inner shape P_I while it is contained in an outer shape P_E. The objects P_I and P_E are assumed to be closed semi-algebraic sets with a total of n boundary features in $\mathbb{R}^d$ for some $d \geq 2$. As a result, the set of acceptable instances of P_M is a family of shapes $S(P_I, P_E) = \{P \subset \mathbb{R}^d | P_I \subseteq P \subseteq P_E\}$ for given P_I and P_E satisfying $P_I \subseteq P_M \subseteq P_E$. In other words, ∂P the boundary of an instance $P \in S(P_I, P_E)$ should be entirely contained in $Q = P_E - int(P_I)$ where $int(P)$ denotes the interior of the set P. The region Q is referred to as the *tolerance zone*. Figure 2.1 shows an example of a planar model part P_M, the inner and outer envelopes P_I and P_E, and a valid instance $P \in S(P_I, P_E)$. We will often refer to a part with shape variation as an *imperfect part*.

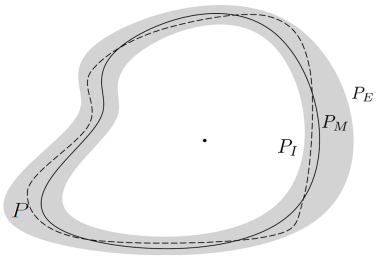

Figure 2.1: A family of shapes specified by an inner envelope P_I and an outer envelope P_E of a model part P_M, along with a valid instance $P \in S(P_I, P_E)$.

2.1.2 Orienting parts with shape variation

We study the problem of orienting a part with shape variations by means of pushing with a line-like friction-less single jaw [70]. See Figure 2.2. Pushing is a mechanically simple type of manipulation and hence is fundamental to many tasks in manipulation.

The basic action of pushing a part at the direction θ consists of placing a single jaw in orientation θ and moving it in a direction perpendicular to itself. When a part P is pushed, it will start a compliant motion (rotation), during which it decreases the distance from its center of mass to the jaw. The motion stops when the normal to the jaw passes through the center of mass of the part. We refer to the corresponding direction of the contact normal as an *equilibrium orientation*. An equilibrium orientation is a *stable orientation* if an edge of the part's convex hull is in contact with the jaw [101].

We define the problem of *orienting an imperfect part* to be that of finding the sequence of push actions that orients the part to the smallest possible orientation set. This possible orientation set consists of disjoint intervals. However, we do not exploit this fact and focus on finding the smallest single interval that contains all possible orientations.

The objects P_I and P_E in this chapter are assumed to be convex and polygonal with a total of n edges. The property of convexity helps us to compute a tight bound on the final orientation of an imperfect part. Also, we assume that the boundaries of P_I and P_E are disjoint.

When there is variation in part shape there will also be variation in the location of the center of mass of the part. In general, the problem of finding the exact locus of the center of mass for a polygon with shape variation is an open problem in [3, 90]. An algorithm for computing a polygonal approximation of the locus will be presened in Chapter 3 under the aforementioned shape variation model. However, for now, we assume that all instances of an imperfect part have their center of mass at the origin. As a result, an instance P

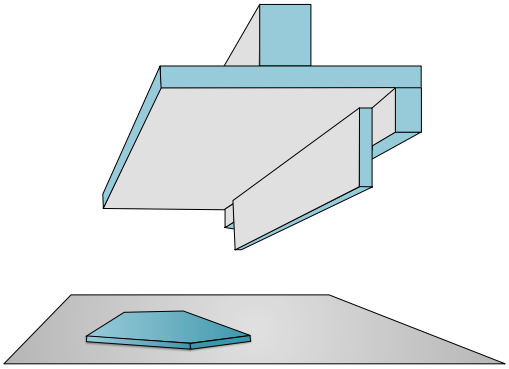

Figure 2.2: A single friction-less jaw can push a planar part resting on a flat surface.

belongs to $S(P_I, P_E)$ if its boundary lies completely inside the tolerance zone Q when its center of mass is placed at the origin.

2.1.3 Definitions for a part

Throughout this chapter, directions are relative to a fixed coordinate frame attached to the origin, increasing in counterclockwise order. Let the set of orientations of P be identified with points on the planar unit circle $S^1 : [0, 2\pi)$. For any orientation θ, the supporting line at the direction θ is a supporting line whose normal vector emanating from the origin has direction θ. See Figure 2.3(b). Pushing P at the direction θ means aligning the jaw with the supporting line at the direction θ. For an interval Θ, we let $L(\Theta)$ and $U(\Theta)$ be the lower and upper bounds (left and right endpoints) of Θ, respectively, and $|\Theta|$ be its length.

The *radius function* $r_P : S^1 \to \mathbb{R}^+$ of a part P maps an angle θ onto the distance between the center of mass and the supporting line of P at the direction θ [101]. The *distance function* $\delta_P : S^1 \to \mathbb{R}^+$ of P maps an angle θ onto the distance between the center of mass and the intersection point of the boundary ∂P of P and the ray emanating from the center of mass at the direction θ [16]. Figure 2.4 depicts the radius functions of P_I and P_E and the distance function of P_E for the illustrated imperfect part. The radius and distance functions are closely related; see Observation 2.1.1.

Observation 2.1.1. *The local minima and maxima of r_P and δ_P coincide; r_P is increasing (decreasing) if and only if δ_P is increasing (decreasing).*

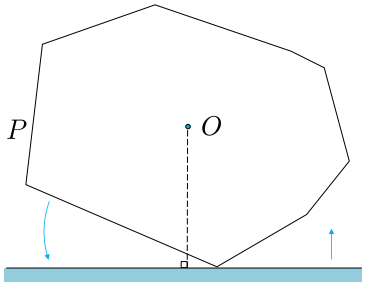

Figure 2.3: A polygon P and its supporting line in the vertical downward direction; when the single jaw moves upward, P rotates in counterclockwise direction.

The *push function* $\phi_P : S^1 \to S^1$ of P maps a push direction of the jaw relative to P in its reference orientation onto the orientation of P after alignment with the jaw. It is well known [70] that the push function follows directly from the radius function as it maps all orientations that are strictly between two consecutive local maxima of the radius function onto the local minimum that is enclosed by these local maxima; moreover, the push function maps each local maximum of the radius function onto itself.

2.1.4 Definitions for a part with shape variation

In this subsection, we define the relevant concepts related to imperfect parts. For simplicity, we use the abbreviations $r_I = r_{P_I}$, $r_E = r_{P_E}$, and $\delta_E = \delta_{P_E}$. Figure 2.4 illustrates an example of an imperfect part and the graph of r_I, r_E and δ_E. The following lemma shows that r_I and r_E bound the radius function of all instances of an imperfect part.

Lemma 2.1.2. *$r_I \leq r_P \leq r_E$ for all $P \in S(P_I, P_E)$.*

Proof. We note that the supporting line of P at θ lies between the supporting lines of P_I and P_E at the same direction. It implies that $r_I \leq r_P \leq r_E$. □

Pushing an imperfect part means pushing an unknown instance from a shape family $S(P_I, P_E)$. As a consequence, the outcome of such a push is the set of all orientations that might result after pushing any shape $P \in S(P_I, P_E)$. To capture this behavior we define the *generalized push function* $\Phi^* : S^1 \to \mathcal{P}(S^1)$, where $\mathcal{P}(S^1)$ denotes the power set of S^1. This function maps an angle θ onto the set of all possible orientations after a single push action in the direction θ, so $\Phi^*(\theta) = \{\phi_P(\theta) | P \in S(P_I, P_E)\}$. As there are several ways to enclose the sets $\Phi^*(\theta)$ by a single interval (due to the cyclic nature of S^1 we must be careful when defining these intervals to avoid ambiguity. To

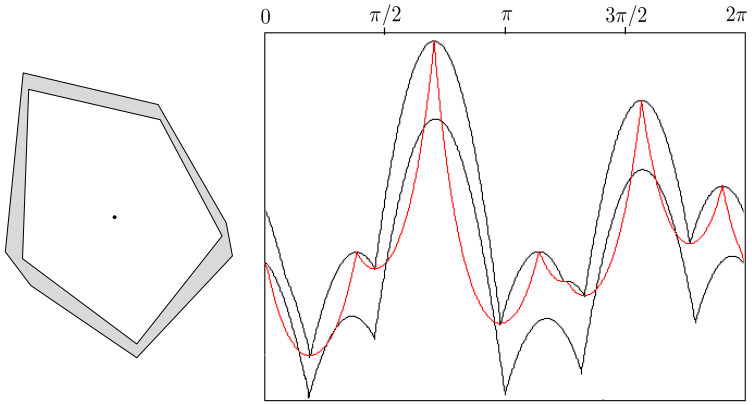

Figure 2.4: An example of an imperfect part, the corresponding graphs of r_I, r_E are illustrated in black and the red graph depicts δ_E.

this end we introduce the *lower push function* and the *upper push function* in Definition 2.1.1.

Definition 2.1.1. *The* lower push function $\Phi_L^* : S^1 \to S^1$ *and* upper push function $\Phi_U^* : S^1 \to S^1$ *are the functions that bound* Φ^* *as follows. We consider three cases based on the push direction* θ*.*

(a) If all instances of $S(P_I, P_E)$ *rotate clockwise when pushed at* θ *then let* α *and* β *be tight upper and lower bounds on the magnitude of the clockwise rotations, respectively. Then* $\Phi_L^*(\theta) = \theta - \alpha$ *and* $\Phi_U^*(\theta) = \theta - \beta$*.*

(b) If all instances of $S(P_I, P_E)$ *rotate counterclockwise when pushed at* θ *then let* α *and* β *be tight lower and upper bounds on the magnitude of the counterclockwise rotations, respectively. Then* $\Phi_L^*(\theta) = \theta + \alpha$ *and* $\Phi_U^*(\theta) = \theta + \beta$*.*

(c) Otherwise let α *and* β *be tight upper bounds on the magnitudes of the clockwise and counterclockwise rotations, respectively. Then* $\Phi_L^*(\theta) = \theta - \alpha$ *and* $\Phi_U^*(\theta) = \theta + \beta$*.*

Note that for each $\theta \in S^1$ the interval $[\Phi_L^*(\theta), \Phi_U^*(\theta)]$ contains the set $\Phi^*(\theta)$. We will denote this interval by $\Phi(\theta)$ and refer to it as the smallest interval containing the set $\Phi^*(\theta)$. Moreover, for an interval $\Theta \subseteq S^1$ we let $\Phi(\Theta) = [\Phi_L^*(L(\Theta)), \Phi_U^*(U(\Theta))]$

We also note that Φ_L^* and Φ_U^* are monotone (non-decreasing), which admits a greedy approach to orient the imperfect part into the smallest possible range of angles. We start with the initial set of possible orientations

$\Theta_0 = [0, 2\pi)$ and repeatedly obtain Θ_{i+1} by selecting it to be the shortest image of any translate of Θ_i under Φ. The process continues as long as $|\Theta_{i+1}| < |\Theta_i|$. To this end, we need to compute the functions Φ_L^* and Φ_U^*. For different types of orientations, the values of these functions are computed differently. These types of angles are defined in the next section.

Note that throughout this chapter, the term right refers to the counterclockwise direction and left refers to the clockwise direction.

Remark. Since range and domain of Φ_L^* and Φ_U^* are S^1, it is possible that $\Phi_L^*(L(\Theta)) > \Phi_U^*(U(\Theta))$. In this case, $|\Phi(\Theta)| = 2\pi + \Phi_U^*(U(\Theta)) - \Phi_L^*(L(\Theta))$.

2.2 Types of orientations

The set of all orientations can be divided into five types based on the computation of their image under Φ_L^* and Φ_U^*. We distinguish two primary types which consist of two and three subtypes respectively.

- An orientation θ is *unstable* if there is no $P \in S(P_I, P_E)$ for which r_P has a local minimum at θ. Such an orientation can never be the final orientation of the imperfect part after pushing. Unstable orientations can be (i) *clockwise unstable*, or (ii) *counterclockwise unstable*.

- An orientation θ is *potentially stable* or *p-stable* if there exists an instance $P \in S(P_I, P_E)$ for which r_P has a local minimum at θ. Such an orientation can be a final orientation of the imperfect part after pushing. Potentially-stable orientations can be (i) *right type (R-type)*, or (ii) *left type (L-type)*, or (iii) *neutral type (N-type)*.

In the following subsections we define the subtypes and properties of p-stable and unstable orientations. The types of orientations divide S^1 into intervals of orientation of the same type. These intervals will be referred to as *critical intervals*. The type of a critical interval equals the type of orientations it contains.

2.2.1 Unstable intervals

Unstable intervals help to reduce the uncertainty in the orientation of an imperfect part as they can never appear in the set of possible orientations after a push action. The following lemma describes how we can distinguish unstable angles.

Lemma 2.2.1. *An orientation $\theta \in S^1$ is unstable if and only if $\delta_E(\theta) < r_I(\theta)$.*

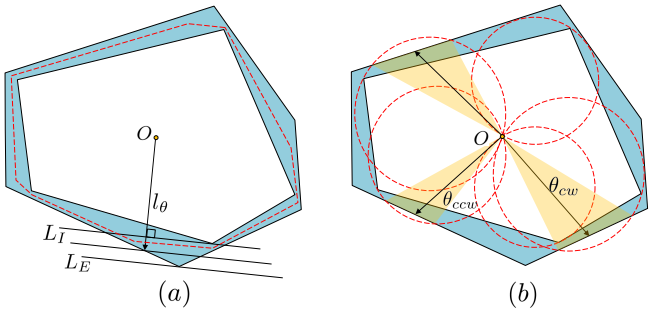

Figure 2.5: (a) θ is an angle such that $\delta_E(\theta) \geq r_I(\theta)$; then there is an instance P such that θ is a local minimum for r_P. (b) Clockwise and counterclockwise unstable intervals can be computed by considering the discs which have a segment connecting the center of mass and a vertex of P_I as its diagonal.

Proof. We first prove the following statement. If $\delta_E(\theta) < r_I(\theta)$, then θ is an unstable angle. Assume for contradiction that θ is a p-stable angle such that $\delta_E(\theta) < r_I(\theta)$. Since θ is p-stable, then there is an instance $P \in S(P_I, P_E)$ such that θ is a local minimum for r_P. According to Observation 2.1.1, $\delta_P(\theta) = r_P(\theta)$ and according to Lemma 2.1.2, $r_I(\theta) \leq r_P(\theta)$. Therefore, $r_I(\theta) \leq \delta_P(\theta)$. We also note that $\delta_P(\theta) \leq \delta_E(\theta)$ implies $r_I(\theta) \leq \delta_E(\theta)$ which is a contradiction.

To prove the reverse implication, let L_I and L_E be the two supporting lines with direction θ of P_I and P_E, respectively. Also, let l_θ be the ray emanating from O in the direction θ. See Figure 2.5(a).

If $\delta_E(\theta) \geq r_I(\theta)$, then the intersection point of ∂P_I and l_θ lies on or between L_I and L_E. So, there is a segment that lies in the tolerance zone which is perpendicular to l_θ. We construct an instance $P \in S(P_I, P_E)$ containing this segment. Since r_P has a local minimum at θ, then θ is a p-stable angle.

□

Figure 2.4 shows several unstable intervals, in which the (red) graph of δ_E lies below the (lower black) graph of r_I. Lemma 2.2.1 shows that we can determine the subdivision of S^1 into unstable and p-stable intervals by computing the intersection of δ_E and r_I.

A geometric way to compute the unstable intervals is to consider the discs that have a segment connecting the center of mass and a vertex of P_I. For any such disc that has at least two intersection points with P_E (resulting in an arc outside of P_E), the angles between direction of the intersection points are unstable. Therefore, any arc outside P_E represents an unstable interval. Figure 2.5(b) illustrates three unstable intervals on an imperfect part. Note that the unstable intervals can be computed in $O(n)$ time since the number

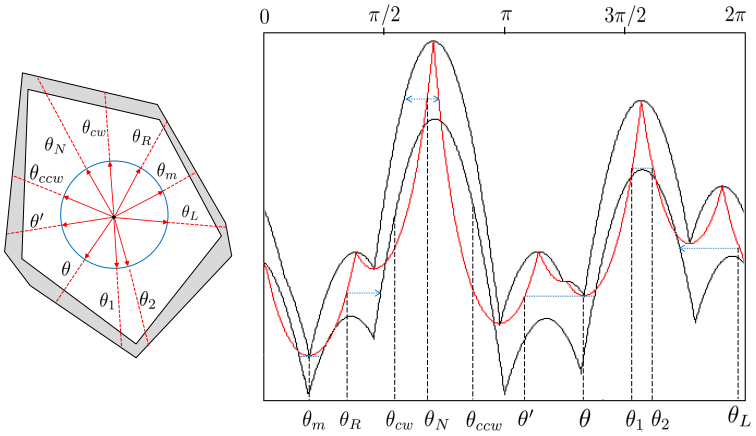

Figure 2.6: The illustrated angles θ_R, θ_L and θ_N are R-type, L-type and N-type, respectively; θ_m is both R-type and L-type. θ_{cw} is a clockwise and θ_{ccw} is a counterclockwise unstable angle. $[\theta', \theta)$ and (θ_1, θ_2) are N-type intervals.

of intersection points cannot exceed $O(n)$.

Observation 2.2.2. *Let $\Theta \subset S^1$ be an unstable interval. All instances $P \in S(P_I, P_E)$ will rotate in the same direction, i.e., in either clockwise or counterclockwise direction, for all push directions $\theta \in \Theta$.*

The above observation shows that there are clockwise and counterclockwise orientations and intervals. For any instance $P \in S(P_I, P_E)$, r_P is strictly increasing in a clockwise unstable interval and strictly decreasing in a counterclockwise unstable interval. In Figure 2.6, θ_{cw} and its containing interval are clockwise unstable while θ_{ccw} and its containing interval are counterclockwise unstable.

2.2.2 Potentially-stable intervals

According to Lemma 2.2.1 p-stable orientations are angles in which the graph of δ_E lies above the graph of r_I. Now consider the graph of δ_E. A p-stable angle θ is called *R-type* if from the point $(\theta, \delta_E(\theta))$ the graph of r_I is horizontally visible to the right. Similarly, it is called *L-type* if the graph of r_I is horizontally visible to the left. If there is no horizontal visibility of r_I the p-stable angle is referred to as *N-type*. In Figure 2.6 the angle θ_R is R-type because the horizontal ray emanating from $(\theta, \delta_E(\theta))$ to the right first hits r_I; θ_L is an L-type angle as the horizontal ray emanating from $(\theta, \delta_E(\theta))$ to the left hits r_I.

The following definition describes the three types of angles more precisely.

Definition 2.2.1. *Let $\theta \in S^1$ be a p-stable angle.*

- *θ is* R-type *if and only if there is no angle ξ such that $\theta < \xi < \theta'$ and $r_E(\xi) = \delta_E(\theta)$, where $\theta' > \theta$ is the smallest angle such that $r_I(\theta') = \delta_E(\theta)$. The angle θ' is called* upper bound *of θ denoted by $B_U(\theta)$.*
- *θ is* L-type *if and only if there is no angle ξ such that $\theta' < \xi < \theta$ and $r_E(\xi) = \delta_E(\theta)$, where $\theta' < \theta$ is the largest angle such that $r_I(\theta') = \delta_E(\theta)$. The angle θ' is called* lower bound *of θ denoted by $B_L(\theta)$.*
- *θ is N-type if it is neither R-type nor L-type.*

The following observation can be made about r_I and r_E. See Figure 2.6.

Observation 2.2.3. *Let $\theta \in S^1$ be R-type (L-type). Then r_E is increasing (decreasing) in a sufficiently small right (left) neighborhood of θ and r_I is increasing (decreasing) in a sufficiently small left (right) neighborhood of $B_U(\theta)$ ($B_L(\theta)$).*

It is possible that an angle is both L-type and R-type. Lemma 2.2.4 shows that such angles are local minima of r_E.

Lemma 2.2.4. *If $\theta \in S^1$ is R-type* and *L-type, then θ is a local minimum of r_E and for all instance $P \in S(P_I, P_E)$, r_P has a local minimum in the interval $[B_L(\theta), B_R(\theta)]$.*

Proof. According to Observation 2.2.3 and considering symmetry, if θ is both R-type and L-type, r_E is increasing in the right neighborhood of θ and decreasing in its left neighborhood. Therefore, θ is a local minimum (like for example θ_m in Figure 2.6). Based on lemma 2.1.2 for all instances $P \in S(P_I, P_E)$, $r_I \leq r_P \leq r_E$. Therefore, the continuous function r_P has two intersection points with the horizontal segment tangent to the graph of r_I at (θ, r_I) with endpoints on $(\theta, B_L(\theta))$ and $(\theta, B_R(\theta))$. According to Roll's theorem, r_P has at least one local minimum in the interval $[B_L(\theta), B_R(\theta)]$.

□

It is not difficult to see that each orientation is of one of the aforementioned types. Lemma 2.2.5 bounds the resulting number of critical intervals and their computation time.

Lemma 2.2.5. *There are $O(n)$ critical intervals; they are computable in $O(n)$ time.*

Proof. To compute the critical intervals, we explore all their potential endpoints. According to Lemma 2.2.1 the intersection points of r_I and δ_E are the open endpoints of clockwise and counterclockwise unstable intervals. The other endpoints must be the critical points for R-type and L-type angles. Recall that by Observation 2.2.3 if θ is R-type (or L-type), then r_E is increasing (decreasing) in the right (left) neighborhood of θ and r_I is increasing (decreasing) in the left (right) neighborhood of $B_U(\theta)$ ($B_L(\theta)$). We also note that if θ is a local minimum or maximum of r_E then $r_E(\theta) = \delta_E(\theta)$, by Observation 2.1.1. Therefore, the remaining endpoints of the critical intervals can be computed by only considering the local minima of r_E and all the local maxima of r_I. We consider the following cases for θ. See Figure 2.6.

- If θ is a local maximum of r_I, then we check whether
  - the horizontal rays emanating from $(\theta, r_I(\theta))$ towards the left and right first hit the graph of δ_E at two other angles θ_1 and θ_2, respectively. In this case, θ_1 is a closed right endpoint of an R-type interval and θ_2 is a closed left endpoint of an L-type interval. In addition, θ_1 and θ_2 are the open left and right endpoint of an N-type interval, respectively.
  - Otherwise, θ does not induce any endpoints.
- If θ is a local minimum of r_E, then we check whether
  - θ is both L-type and R-type. In this case, θ is the closed right endpoint of an L-type interval and closed left endpoint of an R-type interval.
  - θ is only R-type (L-type). In this case, the horizontal ray emanating from $(\theta, r_E(\theta))$ to the right (left) first hits the graph of δ_E at some angle θ'. It can be observed that θ' is the open right (left) endpoint of an R-type (L-type) interval and θ is the closed right (left) endpoint of an R-type (L-type) interval. Moreover, θ and θ' are the two closed endpoints of an N-type interval.
  - θ is N-type, then θ cannot be any endpoint.

Since the number of unstable intervals and the number of local minima and maxima of r_E and r_I are $O(n)$, the number of endpoints of critical intervals cannot exceed $O(n)$. Furthermore, all the endpoints can be computed by a simultaneous walk along the graphs of r_I, r_E, and δ_E in $O(n)$ time. Therefore, the time complexity of computing the critical intervals is $O(n)$.

□

2.3 Computing the lower and upper push functions

To compute Φ_L^* and Φ_U^*, we need to find tight lower and upper bounds for the amount of clockwise or counterclockwise rotation of an imperfect part. (See Definition 2.1.1.) Recall that when a part is pushed, it rotates in the direction in which the radius function decreases. As a result, we are interested in the longest possible non-increasing curve (to the right as well as to the left) that lies completely between r_I and r_E. We note that not every such a curve corresponds to a valid part. Therefore, our strategy is to construct valid instances which create these bounds for clockwise and counterclockwise rotations when they are being pushed at θ.

We will focus on computing upper bounds, Φ_U^*, with the understanding that lower bounds, Φ_L^*, can be computed similarly. In this section, we show that if θ belongs to a clockwise unstable interval then $\Phi_U^*(\theta)$ equals the left endpoint of that interval. If in the other hand, θ belongs to a counterclockwise unstable or a p-stable (L-type, R-type or N-type) interval, then $\Phi_U^*(\theta)$ equals the upper bound of some specific R-type angle.

First we assume that θ is a clockwise unstable angle. So, there is no instance $P \in S(P_I, P_E)$ that rotates counterclockwise. Therefore, the upper bound cannot exceed the left endpoint of the unstable interval that contains θ. This upper bound is easy to compute.

We now assume that θ is not a clockwise unstable angle, so it is a counterclockwise unstable or a p-stable angle. In this case, $\Phi_U^*(\theta) \geq \theta$. We note that if an instance P rotates counterclockwise, then r_P has to be strictly decreasing in a sufficiently small right neighborhood of θ. We define an instance whose radius function is decreasing along the largest possible interval. We refer to this instance as the *upper critical instance* at the direction θ. The critical instance suggests us an approach to compute $\Phi_U^*(\theta)$. We present an algorithm that constructs the upper critical instance for every θ. Then, we prove a theorem that helps to compute Φ_U^* from these critical instances.

By definition, if P is an upper critical instance, then r_P has to be decreasing in the interval $[\theta, \Phi_U^*(\theta)]$. For angles at which r_E is decreasing, the radius function of P_E is decreasing. So, for these angles, we use the corresponding sections of P_E to construct the critical instance. For the other angles we prove the following lemma.

Lemma 2.3.1. *Let $\theta \in S^1$ be an angle such that r_E is increasing in a right neighborhood of θ and let $P \in S(P_I, P_E)$ be an instance that rotates counterclockwise after a single push action at the direction θ. Then $r_P(\theta) \leq \delta_E(\theta)$.*

Proof. Let l_θ be a ray emanating from (the center of mass) O at the direction θ and e be the edge of P_E intersected by l_θ. According to Observation 2.1.1, δ_E is also increasing in the right neighborhood of θ. So, the angle between l_θ and e,

on the left side of l_θ (when walking away from O) and inside P_E is an obtuse angle. See Figure 2.5(a). Let q be the contact point of the supporting line of P at the direction θ. Note that q lies in the tolerance zone Q. Now, assume for contradiction that $r_P(\theta) \geq \delta_E(\theta)$. So, q lies on the left side of l_θ which contradicts the assumption that P rotates in counterclockwise direction when pushed at θ.

□

The next corollary follows from Lemma 2.3.1. Note that r_E and δ_E are increasing in a right neighborhood of an R-type angle.

Corollary 2.3.2. *Let $\theta \in S^1$ be an R-type angle and $P \in S(P_I, P_E)$ be its upper critical instance. Then $r_P(\theta) \leq \delta_E(\theta)$.*

Corollary 2.3.2 reveals that $B_U(\theta)$ is an upper bound on $\Phi^*_U(\theta)$. Note that by Observation 2.2.3 for an R-type angle θ, r_I is increasing in the left neighborhood of $B_U(\theta)$. So, no decreasing curve starting in the right neighborhood of θ can extend beyond $B_U(\theta)$. The following lemma shows that $B_U(\theta)$ is a tight upper bound on $\Phi^*(\theta)$.

Lemma 2.3.3. *Let $d > 0$ be a constant and $[\theta_1, \theta_2] \subset S^1$ be an interval such that for all $\theta \in [\theta_1, \theta_2]$, $r_I(\theta) \leq d \leq \delta_E(\theta)$. Then there is an instance P such that $r_P(\theta) = d$ for all $\theta \in [\theta_1, \theta_2]$.*

Proof. Let ζ be a circular arc with radius of d centered at the center of mass and lying between two rays emanating from the center of mass with directions θ_1 and θ_2. Let P be the object consisting of this circular arc, two line segments tangent to P_I from ζ, and the part of P_I enclosed between them. To show that P is inside the tolerance zone, we must show that ζ is inside the tolerance zone or $\delta_I \leq d \leq \delta_E$. Since $r_I(\theta) \leq d \leq \delta_E(\theta)$ and $\delta_I(\theta) \leq r_I(\theta)$ the arc ζ lies inside the tolerance zone and $r_P(\theta) = d$ for $\theta \in [\theta_1, \theta_2]$.

□

So far, we have discussed how to compute if θ is clockwise unstable or R-type. For the remaining (counterclockwise unstable, L-type and N-type) angles we claim that there is an instance $P \in S(P_I, P_E)$ such that r_P is decreasing in $[\theta, B_U(\theta_m)]$, where θ_m is the closest R-type angle to θ in counterclockwise direction. If such an angle does not exist, then the upper bound is 2π. The following lemma shows that θ_m is a local minimum of r_E.

Lemma 2.3.4. $\Phi^*_U(\theta)$ *If θ is a counterclockwise unstable, L-type or N-type angle, then the closest R-type angle to θ in counterclockwise direction is a local minimum for r_E.*

Proof. Let θ_m be the closest R-type angle to θ to the right. Based on Observation 2.2.3, r_E is increasing in the right neighborhoods of θ_m and r_I is increasing in the left neighborhoods of $B_U(\theta_m)$. Assume for contradiction that θ_m is not a local minimum, r_E has to be increasing in the left neighborhoods of θ_m. Therefore, there exists a sufficiently small $\epsilon > 0$ such that $\theta'_m = \theta_m - \epsilon$ is an R-type angle. It contradicts that θ_m is the closest angle to the right which indicates that the lemma is true.

□

Algorithm 1creates the upper critical instance for an angle θ_0 that is not clockwise unstable. The key idea is that for such an angle θ_0, there is an instance $P \in S(P_I, P_E)$ such that r_P is decreasing in $[\theta_0, B_U(\theta_m)]$ where θ_m is the closest R-type angle to θ_0 in counterclockwise direction.

If there is no such R-type angle, then there is an instance that can rotate arbitrarily close to 2π.

We explain how to construct a decreasing function and then show that this function is a part of the radius function of the instance reported by Algorithm 1. Lemma 2.3.3 shows that any horizontal ray that lies above the graph of r_I and below the graph of δ_E lies on the radius function of some instance. Note that according to Lemma 2.3.1 for any $\theta \in [\theta_0, B_U(\theta_m)]$ if r_E is increasing in the neighborhood of θ and P rotates in counterclockwise direction, then $r_P(\theta) < \delta_E(\theta)$. Therefore, we construct a function for P by starting from θ_0 and follow the horizontal ray emanating from $(\theta_0, \delta_E(\theta_0))$ as long as it stays below δ_E and above r_I. Here P satisfies $r_P(\theta) = \delta_E(\theta_0)$. If the ray hits r_I we are done. Alternatively, it hits δ_E at some angle θ' at which δ_E is decreasing in the right neighborhood of θ'. We continue by choosing $r_P(\theta) = \delta_E(\theta') \cos(\theta' - \theta)$ (which corresponds to a part of the radius function produced by a vertex on the boundary of P_E) until we hit r_E. Then we follow r_E until the closest local minimum and then again we use horizontal rays and continue similarly. The blue graph in Figure 2.7 is an example of a function that is created using this procedure.

Algorithm 1 constructs the corresponding instance which is also shown in Figure 2.7. Note that in our approach we do not actually need to construct the critical instances in order to compute a plan to orient a part. These instances merely show us how to determine $\Phi^*_U(\theta)$ (and $\Phi^*_L(\theta)$).

In Algorithm 1, $P(\theta_1, \theta_2)$ stands for the part of P between two rays emanating from the center of mass in directions θ_1 and θ_2, E_I and E_E are the sets of edges of P_I and P_E respectively, and D_d is the boundary of a disc of radius d centered at the center of mass.

The following lemmas provide the basis for the computation of ϕ^*_U (and ϕ^*_L).

Lemma 2.3.5. *Let $\theta \in S^1$ be R-type and satisfying $\theta = B_U(\theta)$. There is no instance*

Algorithm 1 Constructing the upper critical instance

procedure CONSTRUCT Q(θ_0) ▷ θ_0 is not a clockwise unstable angle
  $Q \leftarrow null$ ▷ Initialization
  $Continue \leftarrow True$
  $d \leftarrow \delta_E(\theta_0)$
  if ∂D_d lies inside the tolerance zone **then**
    $Q \leftarrow D_d$ ▷ Upper critical instance is a disc
  else
    while ($Continue$)
      $d \leftarrow \delta_E(\theta_0)$
      $\theta_1 \leftarrow$ closest angle to θ_0 in counterclockwise
          direction such that ∂D_d intersects the
          segment $s \in E_I \cup E_E$ at direction $\theta_1 \neq \theta_0$
      if $s \in E_I$ **then** ▷ θ_0 is an R-type angle
        $e \leftarrow$ segment on the tangent line of D_d and
            P_I lying between the touching points.
        $\alpha \leftarrow$ direction of the normal vector of e.
        $Q \leftarrow Q \cup D_d(\theta_0, \alpha) \cup e$
        $Continue \leftarrow False$
      else
        if $D_d(\theta_0, \theta_1)$ is inside the tolerance zone **then**
          $Q \leftarrow Q \cup D_d(\theta_0, \theta_1)$
          $\theta_0 \leftarrow \theta_1$
        else
          $\theta_m \leftarrow$ closest local minimum of r_E
              to θ_0 in counterclockwise direction.
          $Q \leftarrow Q \cup P_E(\theta_0, \theta_m)$
          $\theta_0 = \theta_m$
  Construct the rest of P arbitrarily to make it a valid
  instance.
end procedure

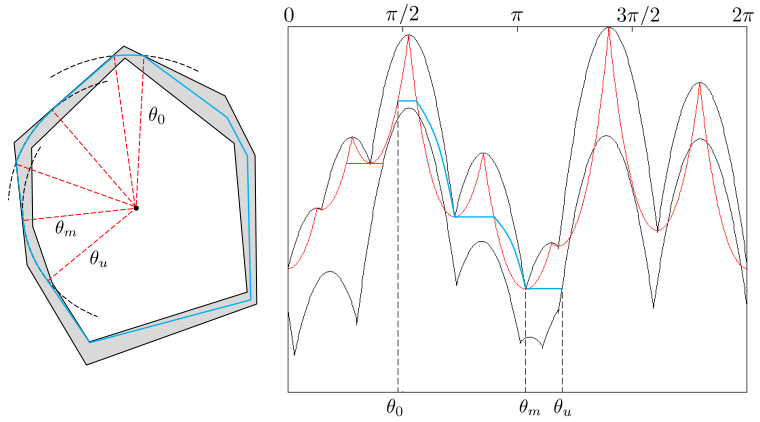

Figure 2.7: Illustration of Algorithm 1 for an imperfect part. The critical instance constructed for the given angle θ_0 is shown in blue. The diagram on the right shows that the corresponding radius function is decreasing; θ_m is the closest R-type angle in counterclockwise direction from θ_0 and $\theta_u = B_U(\theta_m)$.

$P \in S(P_I, P_E)$ that rotates counterclockwise when pushed at θ.

Proof. From $\theta = B_U(\theta)$ it follows that $\delta_E(\theta) = r_I(\theta)$. By Observation 2.2.3, r_I is increasing in the left neighborhood of θ. Assume for a contradiction that r_I is decreasing in the right neighborhood of θ. Then θ is a local maximum of r_I, which corresponds to a vertex. Since $\delta_E(\theta) = r_I(\theta)$, a vertex of P_I has to lie on the boundary of P_E which contradicts the assumption that ∂P_I and ∂P_E are disjoint. So, r_I must be increasing in the right neighborhood of θ. Since $r_I \leq r_P$, the function r_P cannot be decreasing in right neighborhood of θ and therefore P cannot rotate in counterclockwise direction. □

Lemma 2.3.6. *Assume that an imperfect part is pushed at direction θ_0.*

- *If there exists an R-type $\theta \neq \theta_0$, we consider the following cases for the upper bound of the final orientation.*
  - *(a) If θ_0 is a clockwise unstable angle, then the left endpoint θ_u of the containing unstable interval is a tight closed upper bound.*
  - *(b) If θ_0 is not a clockwise unstable angle, then $\theta_u = B_U(\theta_m)$, with θ_m being the closest R-type angle to θ_0 in counterclockwise direction, is a tight open upper bound.*
- *If there is no R-type $\theta \neq \theta_0$ then there is an instance in $S(P_I, P_E)$ that rotates arbitrarily close to 2π.*

Proof. Assume that there exists an R-type $\theta \neq \theta_0$.

(a) θ_0 is a clockwise unstable angle. According to Observation 2.2.2, all the instances rotate in clockwise direction. Since θ_u is an upper bound for p-stable angles to the left of θ_0, there is no instance that can become stable at any orientation larger than θ_u. Also we note that there is an instance such that has a local minimum at θ_0. So, θ_0 is a tight closed upper bound.

(b) θ_0 is not a clockwise unstable angle. Assume for contradiction θ_u is not an upper bound for the final orientation after a push action. So, there is an instance P such that r_P is strictly decreasing in $[\theta_0, \theta_u]$. Therefore, according to Lemma 2.3.5, $\theta_m \neq B_U(\theta_m)$ which implies that $r_P(\theta_u) < r_P(\theta_m)$. According to Lemma 2.1.2, $r_I(\theta_u) \leq r_P(\theta_u)$. Therefore, $r_I(\theta_u) < r_P(\theta_m)$ which implies that $\delta_E(\theta_m) < r_P(\theta_m)$. According to Lemma 2.3.1, P cannot rotate in counterclockwise order at θ_m which is a contradiction. To show that θ_u is open and tight, it is enough to show that for any $\epsilon > 0$ if $\theta_u - \epsilon$ is not a clockwise unstable angle, there is an instance P whose final orientation after a push action at the direction θ_0 is $\theta_u - \epsilon$. We can construct such an instance using the critical instance computed by Algorithm 1. In Algorithm 1, we showed that there is a critical instance such that r_P is (not necessarily strictly) decreasing. It is not difficult to construct an instance sufficiently close to the critical instance between θ_0 and $\theta_u - \epsilon$ such that it lies inside the tolerance zone and does not have any local minimum except at $\theta_u - \epsilon$. Therefore, θ_u is an open tight upper bound.

If there is no R-type $\theta \neq \theta_0$, then according to Lemma 2.3.3 there is a disc centered at O which lies completely in the tolerance zone. So, there is an instance with an infinitely slowly decreasing radius function (almost equal to a constant function). Such an instance rotates arbitrarily close to 2π when is pushed at θ_0

□

Proposition 2.3.7. *The graph of upper push function i.e. Φ^*_U curves upward in an R-type interval and the graph of lower push function i.e. Φ^*_L curves downward in an L-type interval.*

Proof. Note that $r_I(\theta) = d_i \cos(\theta - \theta_i)$ where d_i is the distance form O to the vertex of P_I which touches its supporting at the direction θ and θ_i is the contact direction of that vertex. Also, $\delta_E(\theta) = h_i / \cos(\theta - \theta'_i)$ where h_i is the distance from O to the line passing through the edge of P_E which intersects the ray emanating from O at the direction of θ and θ'_i is the normal direction of that edge. The graph of upper push function i.e. Φ^*_U curves upward in an R-type interval, if the second deravative of Φ^*_U is possitive. Note that by definition for an R-type angle θ, $r_I(\Phi^*_U(\theta)) = \delta_E(\theta)$. Let reverse function of f is denoted by f^{-1}. Then, $\Phi^*_U(\theta) = r_I^{-1}(\delta_E(\theta)) = (\cos^{-1}(h_i / \cos(\theta - \theta'_i)) + \theta_i)/d_i$

for corresponding rang of δ_E and domain of r_I^{-1}. It can be observed that the second deravative of the computed reverse function is possitive. Therefore, the upper push function curves upward in an R-type interval and by symmetry, the lower push function curves downward in an L-type interval.

□

We summarize the discussion of this section in the following theorem.

Theorem 2.3.8. *Φ_L^* and Φ_U^* can be computed in $O(n)$ time.*

Proof. Lemma 2.2.5 shows that the critical intervals can be computed in $O(n)$. For any θ belonging to a critical interval Θ, the function Φ_U^* can be computed by applying Lemma 2.3.6.

- If Θ is clockwise unstable, then $\Phi_U^*(\theta) = L(\Theta)$.
- If Θ is R-type, then $\Phi_U^*(\theta) = B_U(\theta)$. Note that for an R-type angle θ, $r_I(\Phi_U^*(\theta)) = \delta_E(\theta)$. Then, $\Phi_U^*(\theta) = r_I^{-1}(\delta_E(\theta))$ for the corresponding range of δ_E and domain of r_I^{-1}.
- If Θ is counterclockwise unstable, L-type or N-type, then $\Phi_U^*(\theta) = B_U(\theta_m)$ where θ_m is the closest R-type angle in counterclockwise direction. According to Lemma 2.3.4, θ_m is a local minimum of r_E; therefore it is sufficient to check only the local minima. Since the number of local minima is linear and they occur in order, using a simple traversal of the graphs all of them can be computed in a linear time.

The time complexity of computing Φ_U^* is $O(n)$. The same bound applies to Φ_L^*.

□

Figure 2.8(a) shows an example of an imperfect part. The corresponding Φ_L^* and Φ_U^* are depicted by blue and black curves in Figure 2.8(c), respectively. It can be observed that for unstable and N-type intervals, the graphs of Φ_L^* and Φ_U^* are horizontal. For L-type intervals, the graph of Φ_L^* curves downward and the graph of Φ_U^* is horizontal, while for R-type intervals the graph of Φ_U^* curves upward and the graph of Φ_L^* is horizontal.

2.4 An algorithm for orienting an imperfect part

In this section we outline our algorithm for orienting an imperfect part and apply to examples with varying amounts of shape variation.

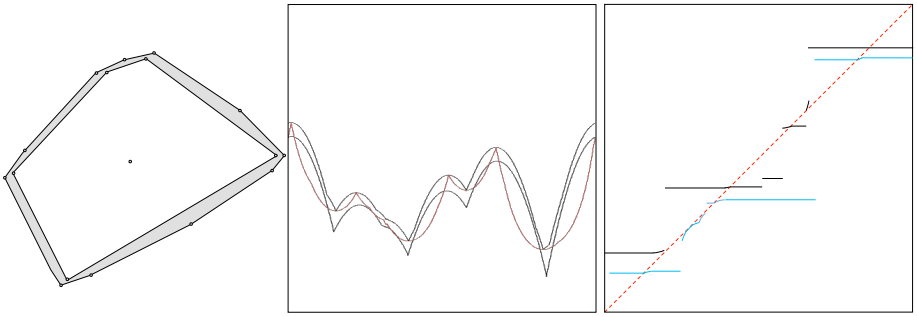

Figure 2.8: (a) an example of an imperfect part. (b) r_I, r_E (black) and δ_E (red). (c) Φ^*_L (blue) and Φ^*_U (black).

2.4.1 A greedy algorithm

In the previous section we have shown how to compute Φ^*_L and Φ^*_U. The monotonicity of these functions admits a greedy approach in the spirit of Goldberg's algorithm [70] to find, for a given integer $h \geq 0$, the sequence of h push actions that orients an imperfect part into the smallest possible interval of orientations. Let Θ_i be the smallest interval containing all possible orientations after i pushes. Obviously, $\Theta_0 = S^1$, and after the first push the part can only be in p-stable orientations, so $\Theta_1 = S^1 - \Pi_{max}$, where Π_{max} is the largest unstable interval. The interval Θ_{i+1} can be obtained by computing the shortest image of any translate of Θ_i under Φ. The process continues as long as $|\Theta_{i+1}| < |\Theta_i|$ and $i < h$. Lemma 2.4.2 helps us to discretize the search for Θ_{i+1} by showing that it suffices to consider only translates of Θ_i in which one of its endpoints coincides with an endpoint of some unstable interval. We first give an observation that is needed to prove Lemma 2.4.2. It says that any p-stable angle θ appears in its own image under Φ^* (and Φ), because, by definition, there is an instance in $S(P_I, P_E)$ that is stable after pushing at θ.

Observation 2.4.1. *$\Phi^*_L(\theta) \leq \theta \leq \Phi^*_U(\theta)$ if and only if θ is a p-stable angle, for any $\theta \in S^1$.*

Lemma 2.4.2. *Let $\Theta \subset S^1$ be an interval with the smallest image under Φ among all the intervals with the length of a given value. If $|\Phi(\Theta)| < |\Theta|$, then there exists an interval $\Theta' \subset S^1$ with $|\Phi(\Theta)| = |\Phi(\Theta')|$ such that $L(\Theta')$ or $U(\Theta')$ coincides with an endpoint of an unstable interval.*

Proof. Assume for contradiction that neither endpoints of Θ are unstable. Since $|\Phi(\Theta)| = |\Phi^*_U(U(\Theta)) - \Phi^*_L(L(\Theta))|$ and $|\Theta| = U(\Theta) - L(\theta)$, according to observation 2.4.1, $|\Phi(\Theta)| \geq |\Theta|$ which is a contradiction. Therefore, at

least one of the endpoint of Θ is unstable. If this endpoint lies on an endpoint of an unstable angle, then the lemma is true. Otherwise, the unstable endpoint of Θ lies on neither endpoints of the corresponding unstable interval. We note that Φ_L^* and Φ_U^* are non-decreasing functions and for an unstable interval they are constant. It means that by shifting the interval to one of the endpoints of the unstable interval we can shrink its image which contradicts the assumption that Θ has the smallest image length or get the same image length. □

Algorithm 2 computes the smallest possible interval of orientations for an imperfect part after (at most) h push actions. Lemma 2.4.2 shows that it suffices to repeatedly align the endpoints of the current smallest interval Θ_i with each of the endpoints of the k unstable intervals Π_j ($1 \leq j \leq k$) to determine Θ_{i+1}.

Algorithm 2 Computing the smallest possible orientation set

procedure COMPUTE-THE-SMALLEST-INTERVAL(Φ_L^*,Φ_U^*,$\Pi = \{\Pi_1,\Pi_2,...,\Pi_k\}$,$h$)
  $i \leftarrow 1$, $X_1 \leftarrow 2\pi - \max_{1\leq j\leq k}\{|\Pi_j|\}$ ▷ Initialization
  Continue $\leftarrow$ *True*
  while (*Continue*) and $i \leq h$
    $S \leftarrow \emptyset$
    for $j = 1$ to k ▷ For all unstable intervals
      $S \leftarrow S \cup \{[L(\Pi_j), L(\Pi_j) + X_i]\} \cup$
      $\{[U(\Pi_j) - X_i, U(\Pi_j)]\}$
    $\Theta_i \leftarrow \Theta \in S$ such that
      $\forall \Theta' \in S, |\Phi(\Theta)| \leq |\Phi(\Theta')|$
    if ($|\Phi(\Theta_i)| < X_i$)
      $i \leftarrow i + 1$
      $X_i \leftarrow |\Phi(\Theta_i)|$
    else
      Continue $\leftarrow$ *False*
  return Θ_i, X_i for all $1 \leq i \leq h$
end procedure

Theorem 2.4.3. *Algorithm 2 finds the sequence of $h \geq 0$ push actions that puts the imperfect part given by $S(P_I, P_E)$ in the smallest interval of possible orientations in $O(hn)$ time.*

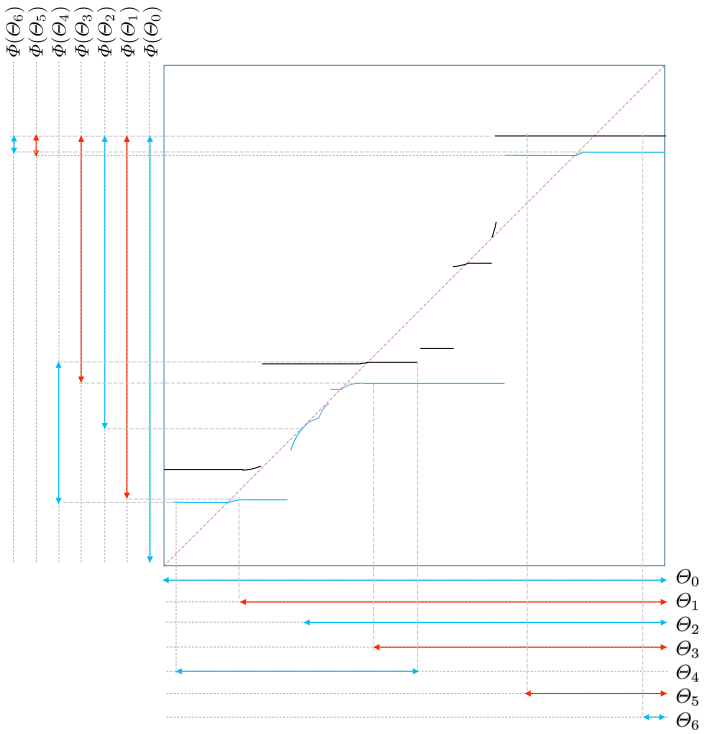

Figure 2.9: Illustration of Algorithm 2 applied to the imperfect part of Figure 2.8(a), showing the intervals Θ_i for $i = 0, \ldots, 8$. The length of the image of any translate of Θ_8 will be at least as long as Θ_8; as a result, no further reduction of the interval of possible orientations is possible.

2.4.2 Applying the algorithm to examples of imperfect parts

In this subsection, we apply Algorithm 2 to several imperfect parts. We first consider the imperfect part depicted in Fig. 2.8(a) and run Algorithm 2 with $h = 10$. Fig. 2.9 illustrates the generalized push function along with the shortest intervals Θ_i of possible orientations after consecutive push actions i. Starting from $\Theta_0 = S^1$, Algorithm 2 terminates after computing Θ_6, which has no translate with an image that is shorter than itself.

It is also interesting to see how different amounts of shape variation and different numbers h of push actions affect the shortest attainable interval Θ_h of possible orientations. We also want to see after how many push actions no further reduction of the interval is possible anymore; we will refer to this number as h^*. We study both phenomena in the context of uniform offsets.

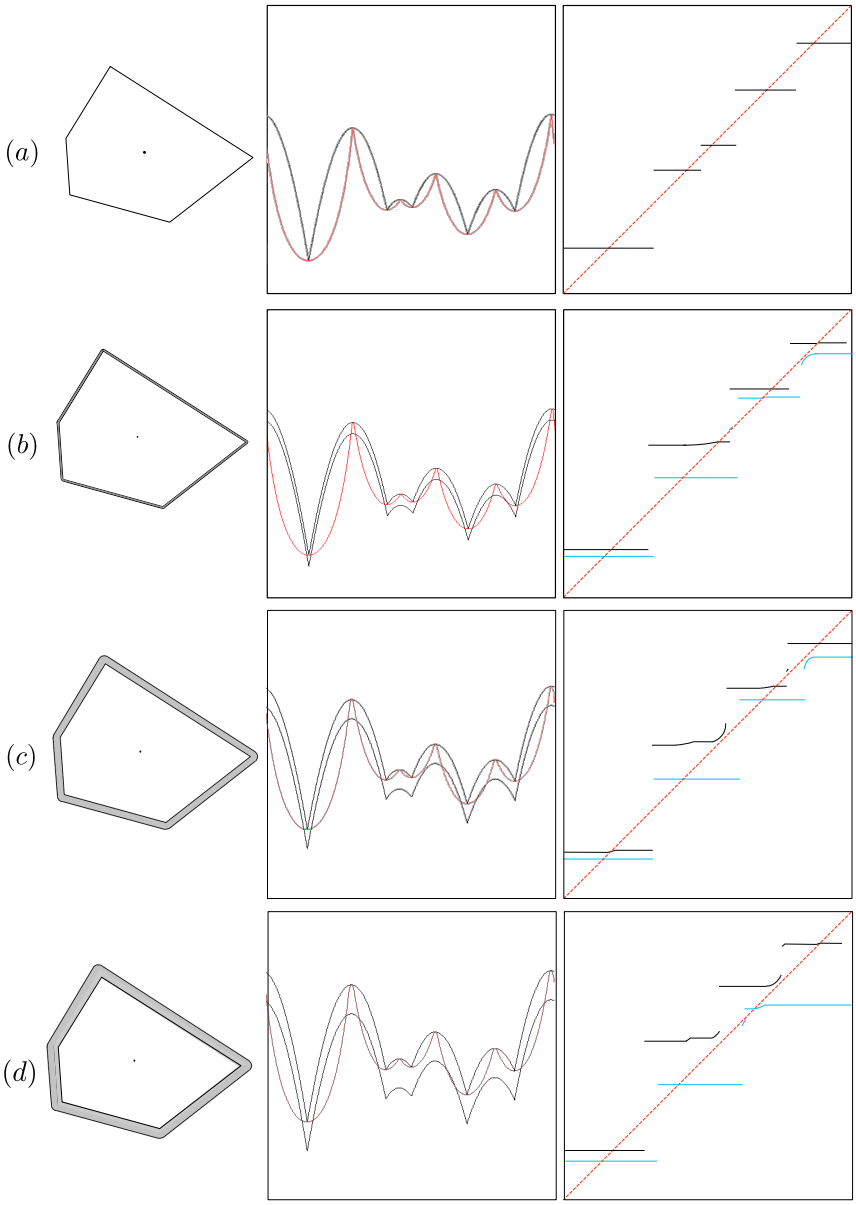

Figure 2.10: The radius and generalized push function for (a) the perfect part e.g. $\sigma = 0$, (b) the imperfect part with variation $\sigma = 0.01$, (c) the imperfect part with variation $\sigma = 0.05$.

We take a polygonal shape P_I of unit diameter and consider different versions of P_E that are obtained by taking the Minkowski sum of P_I with disks of increasing radii σ. More precisely, we let $P_E = P_I \oplus D(\sigma)$, where $D(r)$ is the disk of radius r centered at the origin and $\oplus$ denotes the Minkowski sum operator defined by $A \oplus B = \{a + b | a \in A \wedge b \in B\}$. Notice that the shape of P_E does not differ too much from the shape of P_I—as is normally the case in practice. Fig. 2.10 shows the same P_I and four versions of P_E obtained $\sigma = 0$ (corresponding to a perfect part), $\sigma = 0.01$, $\sigma = 0.02$, and $\sigma = 0.05$ along with the corresponding radius, distance, and upper and lower push functions. It is clear that the size of the gap between Φ^*_U and Φ^*_L increases with the width of the tolerance zone, or, in other words, the distance between the boundaries of P_I and P_E.

Table 2.1: Length of the shortest interval of final orientations for different shape variation σ after applying h push actions.

Offset	Number of push actions					
σ	$h = 3$	$h = 6$	$h = 9$	$h = 12$	$h = 15$	h^*
0.001	199.00°	0.02°				6
0.01	202.30°	2.45°				6
0.02	206.35°	66.15°	5.10°			8
0.03	211.25°	113.70°	7.60°			9
0.04	222.45°	120.90°	10.35°			9
0.05	271.00°	237.45°	153.50°	104.85°	13.15°	15
0.06	274.75°					3
0.07	278.05°					2
0.1	333.45°					1

We have run Algorithm 2 for $\sigma = 0.001, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.1$ and $h = 3, 6, 9, 12, 15$. Table 2.1 displays the interval lengths $|\Theta_h|$ obtained for different combinations of σ and h. Void entries indicate cases in which the algorithm has terminated before h is reached; as a result, the interval length for those entries corresponds the rightmost non-void entry in the same row. The table also displays the number h^* of push actions at which no further reduction of the interval of possible orientations was possible. It can be observed that—as expected—the uncertainty in the final orientation grows if the amount of variation σ increases. The table also shows out that h^* first increases and then decreases when σ increases. It would be interesting to derive upper bounds on h^* as a function of the amount of variation and

the properties of P_I and P_E but it turns out that this is extremely hard, so we pose this as an open problem.

2.5 Optimal orientation set

Instead of running Algorithm 2 for a given maximum number h of pushes, we can also remove that bound and run it as long as the intervals Θ_i continue to shrink, to obtain the largest possible reduction of the uncertainty in the imperfect part's pose. A natural question that arises is whether the algorithm would terminate in that case and thus whether the maximum reduction of pose uncertainty can be obtained after a finite number of pushes. It turns out that it is not always the case. In fact, there are examples in which the final orientation reduces endlessly by applying more number of push actions. In these cases, one can only approach the optimal set of orientations even using infinite pushes. We explore the parts for which the minimum pose uncertainty cannot be obtained after a finite number of pushes. Moreover, we show that for these parts, the optimal set of orientations that one can only approach is computable.

Recall that Algorithm 2 repeatedly aligns the left or right endpoint of an interval Θ_i with one of the $O(n)$ endpoints of an unstable interval Γ. The other endpoint of Θ_i then ends up in one of the $O(n)$ critical intervals, say Γ'. In order to obtain $\Omega(n^2)$ iterations the endpoints of some interval Θ_j with $j > i$ should be able to return to the same pair of intervals consisting of Γ and Γ'.

We assume without loss of generality that the left endpoint of Θ_i (and the future interval Θ_j) coincides with an endpoint of an unstable interval endpoint say Γ_L. The following lemma shows that interval containing the right endpoint(s) say Γ_R must be R-type.

Lemma 2.5.1. *Assume that the left endpoints of two intervals Θ_i and Θ_j in Algorithm 2 for some $j > i$ share the same endpoint of an unstable interval Γ_L and their right endpoints lie in the same critical interval Γ_R. Then Γ_R must be R-type.*

Proof. Assume for contradiction that Γ_R is not R-type. Therefore, Φ_L^* and Φ_U^* are both constant functions in Γ_L and Γ_R. Therefore, the image of any interval whose endpoints are in Γ_L and Γ_R is the same which implies that $|\Phi(\Theta_i)| = |\Phi(\Theta_j)|$. It contradicts the fact that the intervals of the algorithm are getting smaller in each iteration. □

Lemma 2.5.2 helps us to determine the conditions for which the resulting intervals of Algorithm 2 can be shrunk endlessly.

Lemma 2.5.2. *Let $d > 0$ be a constant value and f be a continuous and non-decreasing function which has a derivative at every point in its domain $A \in \mathbb{R}$.*

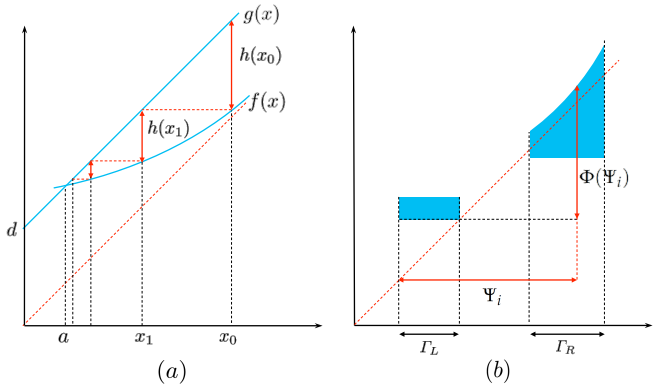

Figure 2.11: (a) The sequence of $x_{n+1} = f(x_n) - d$ where $\lim_{n\to\infty} x_n = a$. (b) $L(\Psi_i)$ lies on $L(\Gamma_L)$ and $U(\Psi_i)$ lies in Γ_R; $|\Phi(\Psi_i)| = |\Psi_i| - |\Gamma_L| + \Phi^*_U(U(\Psi_i)) - U(\Psi_i)$.

Consider the recursive sequence with the general term $x_{n+1} = f(x_n) - d$ and the first element $x_0 > 0$. If this sequence is decreasing and converges to some limit $a \in A$ then

- $f(x) < x + d$, $df/dx > 1$ *where* $x \in [a, x_0]$
- $f(a) = a + d$

Proof. Let $g(x) = x + d$ and $h(x) = g(x) - f(x)$, therefore, $x_{n+1} = x_n - h(x_n) = x_0 - \Sigma_{i=0}^{n} h(x_i)$. Since the sequence is descending, for all $x_n > a$, $h(x_n) = g(x_n) - f(x_n) > 0$ which implies that $f(x) < x + d$ where $x \in [a, x_0]$.

Since $\lim_{n\to\infty} x_n = a$, therefore $\Sigma_{i=0}^{n} h(x_n)$ must be convergent which implies that $\lim_{n\to\infty} h(x_n) = 0$. Since the sequence of x_n is descending, the function $h(x)$ is ascending in $[a, x_0]$. Therefore, $\frac{dh}{dx} > 1$, which implies that $\frac{dh}{dx} = \frac{dg}{dx} - \frac{df}{dx} = 1 - \frac{df}{dx} > 0$. Therefore, $\frac{df}{dx} < 1$. Finally, note that $\lim_{n\to\infty} h(x_n) = 0$, hence $\lim_{n\to\infty} h(a) = g(a) - f(a) = 0$. Therefore, $f(a) = g(a) = a + d$. □

Let Ψ_i $(i > 0)$ be the i-th interval that has its left endpoint in the unstable interval Γ_L and its right endpoint in the critical interval Γ_R. If the number of such intervals is more than one, then according to Lemma 2.5.1, Γ_R is an R-type interval. See Figure 2.11(b).

Considering the identity function and $|\Psi_{i+1}| \leq |\Phi(\Psi_i)|$, it can be observed that $|\Psi_{i+1}| \leq \Phi^*_U(U(\Psi_i)) - \Phi^*_L(L(\Psi_i)) = |\Psi_i| - |\Gamma_L| + [\Phi^*_U(U(\Psi_i)) - U(\Psi_i)]$. Note that $L(\Psi_{i+1}) = L(\Psi_i)$, therefore $U(\Psi_{i+1}) \leq \Phi^*_U(U(\Psi_i)) - |\Gamma_L|$. We also note that $|\Gamma_L|$ is a constant.

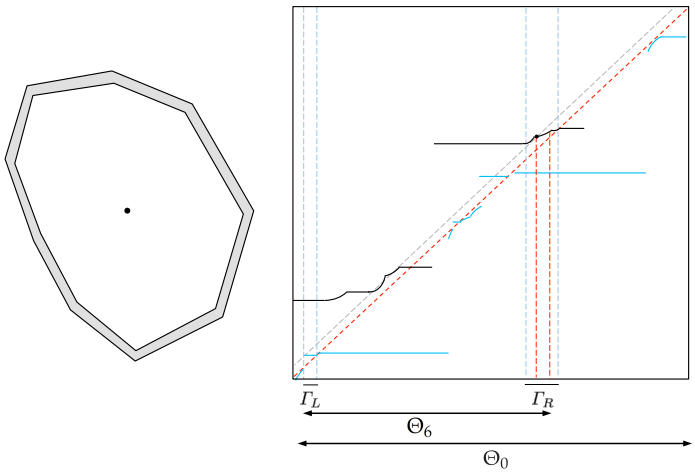

Figure 2.12: An example of an imperfect part and its upper and lower push function. By running the algorithm for $i = 6$, the endpoint of Θ_i lies on Γ_L and Γ_R. It can be observed that Θ_6 satisfies the two conditions. (1) The diagonal gray dashed-line illustrates the graph of $g(\theta) = \theta + |\Gamma_L|$ which intersects the graph of Φ^*_U in Γ_R. (2) Also, it can be observed that $\frac{d\Phi^*_U}{d\theta} < 1$ from the intersection point to $U(\Theta_6)$.

According to lemma 2.5.2 the smallest possible final orientation set can be obtained after a finite number of iterations unless the following condition is met.

There exists an angle $\alpha \in \Gamma_R$ such that:

1. For $\theta \in [\alpha, U(\Psi_i)]$ the graph of $g(\theta) = \theta + |\Gamma_L|$ lies above the graph of Φ^*_U and $d\Phi^*_U/d\theta < 1$.

2. The graph of $g(\theta) = \theta + |\Gamma_L|$ intersects Φ^*_U in α.

According to Lemma 2.5.2, if Ψ_i satisfies both conditions, then the right endpoint of Ψ_i gets close to the intersection point of $g(\theta) = \theta + (|\Gamma_L| - L(\Psi_i))$ and $\Phi^*_U(\theta)$. Therefore, $\lim_{i\to\infty} |\Psi_i| = |\alpha - L(\Gamma_L)|$. So, the final orientation set can get arbitrarily close to this limit by increasing the number of push actions.

Recall that the symmetric case, where Γ_L is L-type, is similar. There exist imperfect parts that meet both of the above conditions for some Θ_i. Figure 2.12 illustrates an example of an imperfect part and its computed lower and upper push function. Running a finite number of iterations of the algorithm on this example cannot result in the optimal set. After 6 steps of running the algorithm the aforementioned condition is met. The diagonal gray dashed-line illustrates the graph of $g(\theta) = \theta + |\Gamma_L|$ which intersects the

graph of Φ^*_U in Γ_R. It can be observed that $\frac{d\Phi^*_U}{d\theta} < 1$ from the intersection point to $U(\Theta_6)$.

2.6 Conclusion

In this chapter, we considered a model for shape variations and studied its effects on orienting parts by pushing. The model allows variations along the entire boundary of a part and between two given closed objects as its inner and outer envelopes. We have proposed an algorithm that takes into account these variations during planning and as such outputs a plan that simultaneously orients all instances satisfying the model into the smallest possible interval of possible orientations after a given number of push actions. For given $h > 0$ and convex polygonal envelopes with n edges in total, the algorithm runs in $O(nh)$ time. We also showed that if we remove the bound on the number of push actions, there are cases that the final orientation set can only approach the optimal set even as the number of pushes goes to infinity. The special cases for which it happens and the optimal set for these cases are computable.

Chapter 3

Location of the center of mass for a part with shape variation

"The world as we have created it is a process of our thinking. It cannot be changed without changing our thinking.

- Albert Einstein

In the previous chapter, we investigated the problem of orienting a part with shape variation using a sequence of pushes. We saw that the shape and the center of mass of a part are crucial parameters to design the algorithms. For simplicity, we assumed that the center of mass is located at the origin for all instances of a part under the proposed model for shape variation. However, the approach presented in the previous chapter can be applied for the case that the center of mass can vary inside a given disk. Note that our approach was based on the radius functions of the inner and outer envelopes and the distance function of the outer envelope. If the center of mass is allowed to vary inside a disk, then the worst cases of these functions can be computed by shifting them vertically by the radius of the disc. Considering the worst cases of the functions and applying the same approach, we can compute an approximation of the smallest final orientation set. This motivated us to study the problem of finding the locus of the center of mass

for all instances of a part with shape variation.

In this chapter, we consider the same model for shape variation and study the relation between variation in part shape and variation in the location of the center of mass for a part with uniform mass distribution. The problem of finding the exact locus of the center of mass is an open problem also for the model that considers variations only in vertices [5, 46]. We determine a polytopic outer approximation for the locus of the center of mass for both two and three dimensional imperfect parts.

For given shpes P_I and P_E such that $P_I \subseteq P_E$ we consider the family of shapes P satisfying $P_I \subseteq P \subseteq P_E$. In the practical setting of toleranced parts P_I and P_E (the inner and outer envelopes) will be fairly similar.

The remainder of the chapter is organized as follows. In the next section, we review the presented model and define the notions that we will use in this chapter. We show in Section 3.2 that the valid instance that yields the largest displacement of the center of mass in a given direction is a shape that combines a part of inner envelope with a part of outer envelope. The corresponding displacement is computable and it can be used to obtain an outer approximation of the set of all possible loci of the center of mass. In Section 3.3, we study the size of the set of possible center-of-mass loci. Fatness of the objects under consideration has led to lower combinatorial complexities and more efficient algorithms for various problems, including union complexities [8], motion planning [131], hidden surface removal [83], and range searching [115]. Here we show that fatness and convexity of P_I together with the assumption that no point in P_E has a distance larger than a given value $\epsilon > 0$ to some point in P_I leads to a bound on the distance between the centers of mass of any two valid instances of a part. This bound is proportional to ϵ and the fatness of P_I. We conclude the chapter in Section 3.4.

3.1 Preliminaries

We review the notion of a center of mass and introduce a few other concepts that allow us to characterize the instances that maximize the displacement of the center of mass. In the presented model in Section 2.1.1, for any manufactured planar model part of P_M, the set of acceptable instances is a family of shapes $S(P_I, P_E) = \{P \subset \mathbb{R}^d | P_I \subseteq P \subseteq P_E\}$ where P_I and P_E are two given closed objects in $\mathbb{R}^d$ satisfying $P_I \subseteq P_M \subseteq P_E$. The closed region resulting from subtracting the interior of P_I from P_E is referred to as the *tolerance zone* and denoted by Q. See Figure 2.3(a). We denote by $COM(P_I, P_E)$ the set of all centers of mass of instances $P \in S(P_I, P_E)$.

We let $X_c(P)$ denote the x-coordinate of the center of mass and $V(P)$ be the volume of the object P, with the understanding that the volume of a two-dimensional object is its area. The x-coordinate of the center of mass of an

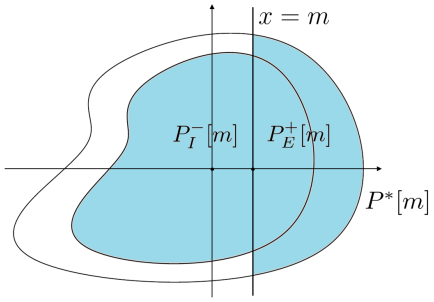

Figure 3.1: A minmax object

object with uniform mass distribution satisfies

$$X_c(P) = \frac{1}{V}\int_V x dV, \tag{3.1}$$

where V is the volume of the object. A similar equality holds for the other coordinate(s) of the center of mass. In the case of uniform mass distribution the center of mass corresponds to the centroid of the object. We will often decompose an object P into sub-objects P_i ($1 \leq i \leq n$) and then express its center of mass as a function of the centers of mass of its constituents, through the equation

$$X_c(P) = \frac{\sum_{i=1}^n X_c(P_i)V(P_i)}{\sum_{i=1}^n V(P_i)}. \tag{3.2}$$

We conclude this section by defining several useful objects. Balls play a prominent role in Section 3.3 of this chapter. We denote by $B(p,r)$ the closed d-dimensional ball with radius r centered at p, and use the abbreviation $B(r) = B(O,r)$ where O is the origin.

For an object P and a value m we define its right portion $P^+[m]$ with respect to m by $P^+[m] = \{(x,y) \in P | x \geq m\}$. Similarly, we define its left portion $P^-[m]$ with respect to m by $P^-[m] = \{(x,y) \in P | x \leq m\}$. With these portions we can define *minmax* objects, which allow us to capture the intuition that the largest displacement of the center of mass in a given direction is achieved by the object from $S(P_I, P_E)$ that 'maximizes mass' in that direction and 'minimizes mass' in the opposite direction. The minmax object $P^*[m]$ consists of a left portion of P_I and a right portion of P_E with respect to the same m, so $P^*[m] = P_I^-[m] \cup P_E^+[m]$ (see Figure 3.1). Note that an alternative way to describe $P^*[m]$ is by the equation $P^*[m] = P_I \cup Q^+[m]$

3.2 Displacement of the center of mass

In this section, we find an upper bound on the displacement of the center of mass in a given direction. The resulting bound allows us to determine a good polytopic outer approximation of the set $COM(P_I, P_E)$ of possible loci of the center of mass.

3.2.1 Bounding the displacement in one direction

Without loss of generality we assume that P_I and P_E are positioned and oriented in such a way that the center of mass of P_I coincides with the origin (so $X_c(P_I) = 0$) and the direction in which we want to bound the displacement aligns with the positive x-axis. Although we will bound the displacement with respect to the center of mass of P_I we observe that the result also induces a bound with respect to the center of mass of P_M as $P_M \in S(P_I, P_E)$ by definition. We let $X_r = \max_{(x,y)\in P_E} x$.

Our first lemma establishes a connection between the minmax objects $P^*[x]$ for $0 \leq x \leq X_r$ and the location of their centers of mass.

Lemma 3.2.1. *There is exactly one minmax object $P^*[m]$ $(0 \leq m \leq X_r)$ that satisfies $X_c(P^*[m]) = m$. Moreover $x < X_c(P^*[x]) \leq m$ for all $0 \leq x < m$ and $X_c(P^*[x]) < m$ for all $m < x \leq X_r$.*

Proof. From $X_c(P_I) = 0$ and $X_c(Q^+[0]) \geq 0$ and the fact that $P^*[0] = P_I \cup Q^+[0]$ it follows that $X_c(P^*[0]) \geq 0$; moreover, it is clear that $X_c(P^*[X_r]) \leq X_r$. As the center of mass of $P^*[x]$ moves continuously as x increases from 0 to X_r there must be at least one x such that $X_c(P^*[x]) = x$. It remains to show that there is also at most one such x. Let m be such that $X_c(P^*[m]) = m$. We consider a minmax object $P^*[x]$ for $x \neq m$ and distinguish two cases: (i) $0 \leq x < m$ and (ii) $m < x \leq X_r$.

Consider case (i). Using the notation $Q' = Q^+[m]$ and $Q'' = P^*[x] - P^*[m] = Q^+[x] - Q^+[m]$ we have that $P^*[m] = P_I \cup Q'$ and $P^*[x] = P_I \cup Q' \cup Q''$. Note that $Q'' \subset [x, m] \times \mathbb{R}$ and thus

$$x \leq X_c(Q'') \leq m.$$

As $x < X_c(P^*[m]) = X_c(P_I \cup Q') = m$ it follows from applying Equation 3.2 to $P^*[m] = P_I \cup Q'$ that

$$x(V(P_I) + V(Q')) < X_c(Q')V(Q') = m(V(P_I) + V(Q')).$$

If we then apply Equation 3.2 to $P^*[x] = P_I \cup Q' \cup Q''$ and use the aforemen-

tioned equation and inequalities we obtain

$$\begin{aligned} X_c(P^*[x]) &= \frac{X_c(Q')V(Q') + X_c(Q'')V(Q'')}{V(P_I) + V(Q') + V(Q'')} \\ &> \frac{x(V(P_I) + V(Q')) + xV(Q'')}{V(P_I) + V(Q') + V(Q'')} = x \end{aligned}$$

and

$$\begin{aligned} X_c(P^*[x]) &= \frac{X_c(Q')V(Q') + X_c(Q'')V(Q'')}{V(P_I) + V(Q') + V(Q'')} \\ &\leq \frac{m(V(P_I) + V(Q')) + mV(Q'')}{V(P_I) + V(Q') + V(Q'')} = m. \end{aligned}$$

Consider case (ii). Using the notation $Q' = Q^+[x]$ and $Q'' = P^*[m] - P^*[x] = Q^+[m] - Q^+[x]$ we have that $P^*[x] = P_I \cup Q'$ and $P^*[m] = P_I \cup Q' \cup Q''$. Note that $Q'' \subset [m, x] \times \mathbb{R}$ and thus

$$m \leq X_c(Q'') \leq x.$$

As $X_c(P^*[m]) = X_c(P_I \cup Q' \cup Q'') = m$ it follows from applying Equation 3.2 to $P^*[m] = P_I \cup Q' \cup Q''$ that

$$X_c(Q')V(Q') = m(V(P_I) + V(Q')) + (m - X_c(Q''))V(Q'').$$

If we then apply Equation 3.2 to $P^*[x] = P_I \cup Q'$ and use the above equations and inequality we obtain

$$\begin{aligned} X_c(P^*[x]) &= \frac{X_c(Q')V(Q')}{V(P_I) + V(Q')} \\ &\leq \frac{m(V(P_I) + V(Q')) - (X_c(Q'') - m)}{V(P_I) + V(Q')} \\ &\leq \frac{m(V(P_I) + V(Q'))}{V(P_I) + V(Q')} = m < x. \end{aligned}$$

Combining both cases we find that there is no $x \neq m$ that satisfies $X_c(P^*[x]) = x$. □

In addition to the fact that there is only one minmax object $P^*[m]$ that satisfies $X_c(P^*[m]) = m$, Lemma 3.2.1 also reveals that $X_c(P^*[x]) > x$ for $x < m$ and $X_c(P^*[x]) < x$ for $x > m$. Moreover, it shows that $X_c(P^*[x]) < m$ for all $x \neq m$ which means that the minmax object $P^*[m]$ with $X_c(P^*[m]) = m$ achieves larger displacement of the center of mass in the direction of the positive x-axis than any other minmax object $P^*[x]$ with $x \neq m$. The following theorem shows that $P^*[m]$ in fact achieves the largest displacement of the center of mass among *all* objects in $S(P_I, P_E)$.

Theorem 3.2.2. *Let $P^*[m]$ $(0 \leq m \leq X_r)$ be the unique minmax object that satisfies $X_c(P^*[m]) = m$. Then $X_c(P) < X_c(P^*[m])$ for all $P \in S(P_I, P_E), P \neq P^*[m]$.*

Proof. Let $P \in S(P_I, P_E), P \neq P^*[m]$ be the object that yields the largest displacement $m' \geq m$ of the center of mass, so $X_c(P) = m'$. If $P = P^*[m']$ then it follows immediately from Lemma 3.2.1 that $m' = m$. Now assume for a contradiction that $P \neq P^*[m'] = P_I^-[m'] \cup P_E^+[m']$ which implies that (i) $P_E^+[m'] - P^+[m'] \neq \emptyset$ or (ii) $P^-[m'] - P_I^-[m'] \neq \emptyset$.

Consider case (i) and let R be a closed connected subset with $V(R) > 0$ of $P_E^+[m'] - P^+[m']$. Observe that $P \cup R \in S(P_I, P_E)$. Note that $R \subset (m', \infty) \times \mathbb{R}$ and thus $X_c(R) > m'$. We get

$$\begin{aligned} X_c(P \cup R) &= \frac{X_c(P)V(P) + X_c(R)V(R)}{V(P) + V(R)} \\ &> \frac{m'V(P) + m'V(R)}{V(P) + V(R)} = m' \end{aligned}$$

which contradicts the assumption that P is the object in $S(P_I, P_E)$ that achieves the largest displacement of the center of mass.

Consider case (ii) and let R be a closed connected subset with $V(R) > 0$ of $P^-[m'] - P_I^-[m']$. Observe that $P - R \in S(P_I, P_E)$. Note that $R \subset (-\infty, m') \times \mathbb{R}$ and thus $X_c(R) < m'$. We get

$$\begin{aligned} X_c(P - R) &= \frac{X_c(P)V(P) - X_c(R)V(R)}{V(P) - V(R)} \\ &> \frac{m'V(P) - m'V(R)}{V(P) - V(R)} = m \end{aligned}$$

which again contradicts the assumption that P is the object in $S(P_I, P_E)$ that achieves the largest displacement of the center of mass. As a result we find that $P^*[m]$ with $X_c(P^*[m]) = m$ is the unique object in $S(P_I, P_E)$ that achieves the largest displacement of the center of mass. □

The theorem shows that the set $COM(P_I, P_E)$ does not extend beyond the plane or line $x = m$ where m is such that $X_c(P^*[m]) = m$. The bound is tight because $P^*[m] \in S(P_I, P_E)$. In fact, the theorem shows that $P^*[m]$ is the only instance in $S(P_I, P_E)$ that has its center of mass on that plane or line. Since the result holds in any direction, this implies that $COM(P_I, P_E)$ is convex.

3.2.2 A k-facet approximation for $COM(P_I, P_E)$

The results in the previous subsection suggest an easy approach to determine an outer approximation of the set $COM(P_I, P_E)$ of possible centers of mass

of instances in $S(P_I, P_E)$. If we select k different directions that positively span the d-dimensional space ($d = 2, 3$) and apply Theorem 3.2.2 in each of these directions then we obtain a bounded polytope with k facets enclosing $COM(P_I, P_E)$. Every facet of the polytope contains a point of the convex set $COM(P_I, P_E)$.

Our method to efficiently compute the largest displacement of the center of mass in the positive x-direction relies on a covering P_I and P_E by $O(n)$ signed cones, following an idea by Lien and Kajiya [93]. To cover P_I we pick a point v on its boundary and decompose. We confine ourselves to the case where P_I and P_E are convex as this allows us to efficiently decompose both shapes into a $O(n)$ convex pieces of constant complexity, e.g, by first subdividing the boundary patches into constant-complexity subpatches and then connecting all these subpatches to a single vertex. To find the largest displacement of the center of mass in the positive x-direction, we sort the vertices of P_I and P_E by x-coordinate and perform a binary search. For each x considered during this search we use the intersections of the constant-complexity pieces of P_I and P_E with the respective half-planes defined by x to compute $X_c(P^*[x]) = X_c(P_I^-[x] \cup P_E^+[x])$ in $O(n)$ time, which is then compared to x itself to determine how to continue the search. Once the m satisfying $X_c(P^*[m]) = m$ is found to lie between two consecutive vertices, its value can be computed in $O(n)$ time.

Theorem 3.2.3. *A polytopic k-facet outer approximation of $COM(P_I, P_E)$ can be computed in $O(kn \log n)$ time if P_I and P_E are convex.*

Figure 3.2 illustrates 4-, 8-, 16-, and 64-vertex outer approximations of $COM(P_I, P_E)$ for a given P_I and P_E. Recall that every edge of the polygonal approximation contains one point of the convex set $COM(P_I, P_E)$, so $COM(P_I, P_E)$ strongly resembles its approximation.

The examples in Figure 3.2 seem to suggest that the displacement of the center of mass is proportional to the distance between the boundaries of P_I and P_E and does not depend on the sizes of P_I and P_E themselves. In the next section we will see that this is not true in general. We will derive a bound on the size of $COM(P_I, P_E)$ for a convex P_I that depends on the distance between the boundaries of P_I and P_E and the fatness of P_I.

3.3 Bounding the size of $COM(P_I, P_E)$

The admitted shape variation for a manufactured part is usually small compared to the dimensions of the part itself. As a result, the enclosed shape P_I and enclosing shape P_E do not deviate much from the model shape P_M, and therefore also not from each other. To capture this similarity we will assume that $P_I \subseteq P_E \subseteq P_I \oplus B(\epsilon)$, where $\oplus$ denotes the Minkowski sum. Note that

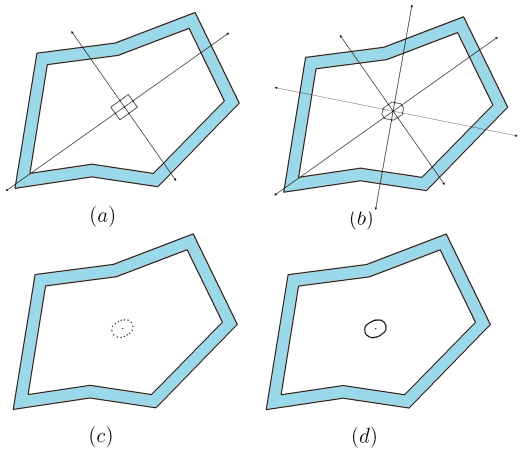

Figure 3.2: Outer approximations of $COM(P_I, P_E)$ with (a) 4, (b) 8, (c) 16, and (d) 64 vertices.

this means that every point in P_E lies within a distance of at most ϵ from some point in P_I.

We must also assume that P_I is convex and fat to obtain a bound on the diameter of $COM(P_I, P_E)$ that depends on ϵ and the fatness. There are many different definitions of fatness and we will use the one by de Berg and van der Stappen [55], which is based on a similar definition presented in the thesis of van der Stappen [131].

Definition 3.3.1. *Let $P \subseteq \mathbb{R}^d$ be an object and let β be a constant with $0 < \beta \leq 1$. Define $U(P)$ as the set of all balls centered inside P whose boundary intersects P. We say that the object P is β-fat if for all balls $B \in U(P)$ we have $V(P \cap B) \geq \beta \cdot V(B)$. The fatness of P is defined as the maximal β for which P is β-fat.*

For bounded objects the value of β is at most $1/2^d$; larger values only occur for unbounded objects [131].

Two planar polygonal examples in Subsections 3.3.1 and 3.3.2 show that both fatness and convexity of P_I are necessary for a bound that is independent of the size of P_I (and P_E). In Subsection 3.3.3 we derive a bound for the case that both assumptions hold.

3.3.1 A thin convex part

When P_I is a sufficiently long and narrow box the set $S(P_I, P_E)$ contains shapes whose centers of mass are a distance proportional to the diameter of P_I apart. Let $L \gg \epsilon$ and pick λ such that $0 < \lambda < 2\epsilon^2/(L - \epsilon)$. We define

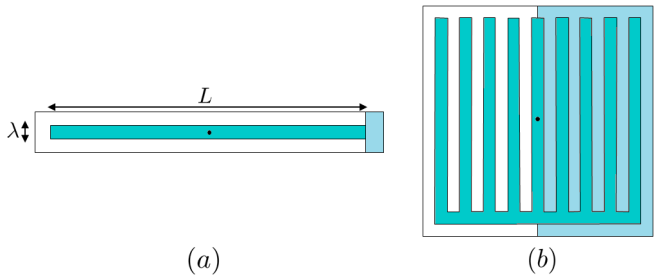

Figure 3.3: (a) A thin convex part (b) A fat non-convex part

$P_I = [-L/2, L/2] \times [-\lambda/2, \lambda/2]$ and $P_E = [-(L/2+\epsilon), L/2+\epsilon] \times [-(\lambda/2+\epsilon), \lambda/2+\epsilon]$, and note that $P_E \subseteq P_I \oplus B(\epsilon)$. See figure 3.3$(a)$.

Now consider the object $P^*[L/2] = P_I^-[L/2] \cup P_E^+[L/2] = P_I \cup P_E^+[L/2]$. We observe that $V(P_I) = \lambda L$, $X_c(P_I) = 0$, $V(P_E^+[L/2]) = \epsilon(2\epsilon + \lambda)$, and $X_c(P_E^+[L/2]) = L/2 + \epsilon/2 > L/2$. The upper bound on λ implies that $V(P_E^+[L/2]) > V(P_I)$. From Equation 3.2 it follows that $X_c(P^*[L/2]) > L/4$ showing that the diameter of $COM(P_I, P_E)$ is *not* proportional to ϵ in this case.

3.3.2 A fat non-convex part

Let $P_E = [-L/2, L/2] \times [-L/2, L/2]$, and consider a box $[-L/2+\epsilon, L/2+\epsilon] \times [-L/2+\epsilon, L/2+\epsilon]$ and subdivide it both horizontally and vertically into an odd number (≥ 5) of strips of width smaller than ϵ. We construct P_I by taking every second vertical strip, starting with the first and ending with the last, and the bottommost horizontal strip. The union of all these strips is a comb-shaped object that is known to be at least $1/4\pi$-fat [131]. From the symmetry of P_I it is immediately clear that $X_c(P_I) = 0$.

Now consider the object $P^*[0] = P_I \cup Q^+[0]$. It is clear from the construction that $V(Q^+[0]) > V(P_I)/2$. It is also easy to verify that $X_c(Q^+[0]) > L/4$. Combining the inequalities with Equation 3.2 yields that $X_c(P^*[0]) > L/16$ showing that the diameter of $COM(P_I, P_E)$ is also *not* proportional to ϵ in this case.

3.3.3 Fat convex parts

We turn our attention to the case that P_I is both convex and β-fat ($0 < \beta \leq 1$), and recall that $P_E \subseteq P_I \oplus B(\epsilon)$. The Steiner formula for convex bodies (see e.g. [11, 107]) establishes a useful connection between the properties of a convex object P and a value ϵ on the one hand and the volume of $P \oplus B(\epsilon)$

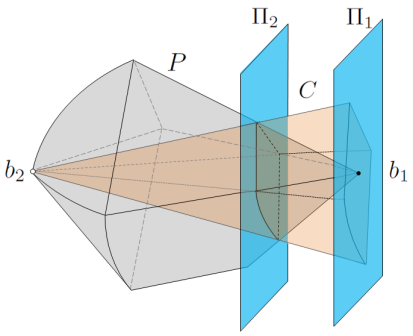

Figure 3.4: Illustration of Lemma 3.3.2.

on the other hand. Lemma 3.3.1 summarizes the formula for two- and three-dimensional objects.

Lemma 3.3.1. *Let $P \subset \mathbb{R}^d$ $(d = 2, 3)$ be a convex object. Then*

- $V(P \oplus B(\epsilon)) = V(P) + p\epsilon + \pi\epsilon^2$, *where p is the perimeter of P, when $d = 2$, and*
- $V(P \oplus B(\epsilon)) = V(P) + a\epsilon + 2\pi w\epsilon^2 + \frac{4}{3}\pi\epsilon^3$, *where a is the surface area and w is the mean width*[1] *of P, when $d = 3$.*

The results will be used in the proof of Theorem 3.3.3 to bound the additional volume that any instance of $S(P_I, P_E)$ can have in comparison to P_I.

Lemma 3.3.2 is not strictly necessary yet it leads to a better bound in our main theorem. The two-dimensional version of the lemma is based on a result by Hammer [77], which we will generalize to three-dimensional objects.

Lemma 3.3.2. *Let $P \subset \mathbb{R}^d$ $(d = 2, 3)$ be a convex object with diameter δ. Then no point in P has distance larger than $\frac{\delta d}{d+1}$ to the center of mass of P.*

Proof. Hammer's result [77] says that the center of mass of every planar convex body divides every chord through it in a ratio less than or equal to 2/3, meaning that the part of the chord on one side of the center of mass cannot

[1] In geometry, the *mean width* is a measure of for the *size* of a body. The *width* of a 3D object in a given direction $\hat{n}$ is the distance between the parallel planes with normal $\hat{n}$ tangent to the object. The mean width is the average of this width over all $\hat{n}$ in S^2 where S^2 is the surface of a 3D sphere.

be more than twice as long as the part on the other side of the center of mass. The result then follows from the observation that no chord is longer than the diameter δ.

We will extend Hammer's construction to show that the center of mass of every three-dimensional convex body divides every chord through it in a ratio less than or equal to $3/4$, which will then immediately imply the given result. Let b_1 and b_2 be the endpoints of a chord through the center of mass of P. See figure 3.4 Let Π_1 be a plane tangent to P at b_1, and let Π_2 be the plane parallel to Π_1 at $3/4$ the distance between b_2 and Π_1 from b_2. The intersection of P with the plane Π_2 is a convex two-dimensional shape I. We create a generalized cone C by taking the union of all half-lines emanating from b_2 and passing through I and clipping the resulting shape with the plane Π_1. Application of Equation 3.1 to C reveals that its center of mass lies on Π_2.

The plane Π_2 cuts the objects P and C into two parts each. Let P_1 and C_1 be the parts of P and C respectively between Π_1 and Π_2; let P_2 and C_2 be the parts of P and C respectively in the half-space bounded by Π_2 an containing b_2. The convexity of P implies that $P_1 \subseteq C_1$ and $P_2 \supseteq C_2$; in other words, the object P has less or equal mass than C beyond Π_2, while it has more or equal mass than C in front of Π_2 (when viewed from b_2). As a result, the object P must have its center of mass on the part of the chord between b_1 and b_2 in front of Π_2, which proves the claim. □

Lemma 3.3.2 shows that any convex object with diameter δ (and uniform mass distribution) fits completely inside a ball with radius $\frac{\delta d}{d+1}$ centered at its center of mass, which is a slightly stronger result than the obvious claim that it fits inside a ball with radius δ.

We now have all the ingredients to prove our upper bound on the diameter of $COM(P_I, P_E)$.

Theorem 3.3.3. *Let $P_I \subset \mathbb{R}^d$ ($d = 2, 3$) be a bounded convex β-fat object ($0 < \beta \leq 1$) and let $P_E \subset \mathbb{R}^d$ be a bounded object satisfying $P_I \subseteq P_E \subseteq P_I \oplus B(\epsilon)$. Then the diameter of $COM(P_I, P_E)$ is bounded by $\frac{5}{2}\beta^{-1}\epsilon$ if $d = 2$ and by $3\beta^{-1}\epsilon$ if $d = 3$.*

Proof. We use δ to denote the diameter of P_I and once again assume without loss of generality that $X_c(P_I) = 0$. Theorem 3.2.2 shows that it suffices to consider objects $P^*[m]$ to bound the size of $COM(P_I, P_E)$. The assumption $X_c(P_I) = 0$ allows us to simplify Equation 3.2 for $P^*[m] = P_I \cup Q^+[m]$ to

$$X_c(P^*[m]) = \frac{X_c(Q^+[m])V(Q^+[m])}{V(P_I) + V(Q^+[m])}. \tag{3.3}$$

Lemma 3.3.2 says that P_I lies completely inside $B(\delta d/(d+1))$. As a consequence, the object $P^*[m]$ must lie entirely inside $B(\delta d/(d+1) + \epsilon)$, which

implies that $X_c(P^*[m]), X_c(Q^+[m]) \leq (\delta d/(d+1) + \epsilon)$. We treat $d = 2$ and $d = 3$ separately.

Consider $d = 2$. We distinguish two cases based on the ratio of ϵ and δ.

If $\epsilon \geq \delta/6$ then $X_c(P^*[m]) \leq 2\delta/3 + \epsilon \leq 5\epsilon$. Since $P^*[m]$ is bounded we know that $\beta \leq 1/4$ and thus $X_c(P^*[m]) \leq 5\epsilon \leq 5\beta^{-1}\epsilon/4$.

If $\epsilon \leq \delta/6$ we use Equation 3.3 to obtain an upper bound $X_c(P^*[m])$ by combining the upper bound $X_c(Q^+[m]) \leq 2\delta/3 + \epsilon$ with a lower bound on $V(P_I)$ and upper and lower bounds on $V(Q^+[m])$. The lower bound on $V(P_I)$ follows from the fatness of P_I. As δ is the diameter of P_I there must be two points $p_1, p_2 \in P_I$ that are δ apart. The boundary of the ball $B(p_1, \delta)$ contains p_2 and thus belongs to the set $U(P_I)$. The β-fatness of P_I implies that $V(P_I) \geq \beta \cdot V(B(p_1, \delta)) = \beta\pi\delta^2$. It remains to bound $V(Q^+[m])$. We note that $Q^+[m] \subseteq Q \subseteq (P_I \oplus B(\epsilon)) - int(P_I)$, from which it follows that $V(Q^+[m]) \leq V(P_I \oplus B(\epsilon)) - V(P_I)$. Lemma 3.3.1 says that $V(P_I \oplus B(\epsilon)) - V(P_I) = p\epsilon + \pi\epsilon^2$, where p is the perimeter of P. As P_I is contained in $B(2\delta/3)$ we know that $p \leq 4\pi\delta/3$. Combining these observations with a trivial lower bound on $V(Q^+[m])$ we get $0 \leq V(Q^+[m]) \leq 4\pi\epsilon\delta/3 + \pi\epsilon^2$.

Plugging all the inequalities into Equation 3.3 and using $\epsilon/\delta \leq 1/6$ yields

$$\begin{aligned} X_c(P^*[m]) &= \frac{X_c(Q^+[m])V(Q^+[m])}{V(P_I) + V(Q^+[m])} \\ &\leq \frac{(\frac{2}{3}\delta + \epsilon)(\frac{4}{3}\pi\delta\epsilon + \pi\epsilon^2)}{\beta\pi\delta^2} \\ &= \beta^{-1}\epsilon(\tfrac{8}{9} + 2(\tfrac{\epsilon}{\delta}) + (\tfrac{\epsilon}{\delta})^2) \\ &\leq \tfrac{5}{4}\beta^{-1}\epsilon. \end{aligned}$$

which shows $COM(P_I, P_E) \subseteq B(\frac{5}{4}\beta^{-1}\epsilon)$ if $d = 2$.

Now consider $d = 3$, and again distinguish two cases based on the ratio of ϵ and δ. [2]

If $\epsilon \geq 3\delta/44$ then $X_c(P^*[m]) \leq 3\delta/4 + \epsilon \leq 12\epsilon$. As $P^*[m]$ is bounded we have that $\beta \leq 1/8$ and thus $X_c(P^*[m]) \leq 12\epsilon \leq 3\beta^{-1}\epsilon/2$.

If $\epsilon \leq 3\delta/44$ we again derive a lower bound on $V(P_I)$ and upper and lower bounds on $V(Q^+[m])$, and then combine it with the upper bound $X_c(Q^+[m]) \leq 3\delta/4 + \epsilon$. The lower bound on $V(P_I)$ follows from the fatness of P_I. If $p_1, p_2 \in P_I$ are δ apart, then the ball $B(p_1, \delta)$ again belongs to the set $U(P_I)$. The β-fatness of P_I now implies that $V(P_I) \geq \beta \cdot V(B(p_1, \delta)) = 4\beta\pi\delta^3$.

[2] For reasons of simplicity of the final bound we have chosen the split at $\epsilon = \delta/(44/3)$. The optimal split would be at $\epsilon = \delta/k$ where k equals the single positive real root of the equation $6k^4 - 73k^3 - 180k^2 - 144k - 64 = 0$, which yields

$$k = \sqrt[3]{\tfrac{977}{8} + \sqrt{654}} + \sqrt[3]{\tfrac{977}{8} + \sqrt{654}} + \tfrac{9}{2}.$$

A split at k leads to a marginally better bound on the radius of the disk containing $COM(P_I, P_E)$.

Now we bound $V(Q^+[m])$ using $V(Q^+[m]) \leq V(P_I \oplus B(\epsilon)) - V(P_I)$. Lemma 3.3.1 says that $V(P_I \oplus B(\epsilon)) - V(P_I) = a\epsilon + 2\pi w\epsilon^2 + \frac{4}{3}\pi\epsilon^3$, where a is the surface area and w is the mean width of P. As P_I is contained in $B(3\delta/4)$ we know that $a \leq 4\pi(3\delta/4)^2 = 9\pi\delta^2/4$. Moreover, the mean width w does not exceed the diameter of P_I so $w \leq \delta$. Combining these observations with a trivial lower bound on $V(Q^+[m])$ we get $0 \leq V(Q^+[m]) \leq 9\pi\delta^2\epsilon/4 + 2\pi\delta\epsilon^2 + 4\pi\epsilon^3/3$.

Plugging all the inequalities into Equation 3.3 and using $\epsilon/\delta \leq 3/44$ yields

$$\begin{aligned} X_c(P^*[m]) &= \frac{X_c(Q^+[m])V(Q^+[m])}{V(P_I) + V(Q^+[m])} \\ &\leq \frac{(\frac{3}{4}\delta + \epsilon)(\frac{9}{4}\pi\delta^2\epsilon + 2\pi\delta\epsilon^2 + \frac{4}{3}\pi\epsilon^3)}{\frac{4}{3}\beta\pi\delta^3} \\ &= \beta^{-1}\epsilon(\tfrac{81}{64} + \tfrac{45}{16}(\tfrac{\epsilon}{\delta}) + \tfrac{9}{4}(\tfrac{\epsilon}{\delta})^2 + (\tfrac{\epsilon}{\delta})^3) \\ &< \tfrac{3}{2}\beta^{-1}\epsilon. \end{aligned}$$

which shows $COM(P_I, P_E) \subseteq B(\frac{3}{2}\beta^{-1}\epsilon)$ if $d = 3$. □

Theorem 3.3.3 confirms the intuition that the variation of the center of mass of a convex part grows if the admitted shape variation increases or the fatness decreases.

3.4 Conclusion

We have considered a general model for admitted shape variations of a model part, based on an enclosed convex shape P_I and an enclosing convex shape P_E in both two and three dimensional space. We have focused on two and three dimensional space and identified the valid instance that maximizes the displacement of the center of mass in a given direction. We used this result to find a k-vertex polytopic outer approximation of the set of all possible center-of-mass loci in $O(kn)$ time, where n is the number of vertices of P_I and P_E. If P_I is β-fat and every point of P_E is within a distance ϵ of P_I then the diameter of the set of all center-of-mass loci can be shown to be $O(\beta^{-1}\epsilon)$.

Chapter 4

Pose statistics for eccentric parts

"The most important questions of life are indeed, for the most part, really only problems of probability."

- Pierre Simon Laplace

FOR many automated manufacturing tasks such as part orienting, it is very useful to have an estimate of the final orientation of a part which is dropped onto a flat surface. The shape of a part is usually known, but its starting orientation or location is not. As mentioned in Chapter 1 the basic objective of part feeders is to minimize the uncertainty in the orientation of the part before the next stage of the manufacturing process. This is often achieved by applying a sequence of actions regardless of initial orientation. In this chapter, we consider the final orientation of an object initially at a random orientation, and show how the shape of the object relates to the distribution of its final orientation.

For some parts, it is possible to predict that the poses into which they settle with high probability will be in a cluster of poses which are very close together. Considering these poses as the most probable initial orientations of the part for feeding allows faster design for part feeding tasks. Observing the usefulness of bias in pose distribution for part feeding, we consider a variant of the notion of *geometric eccentricity* for objects and show that there

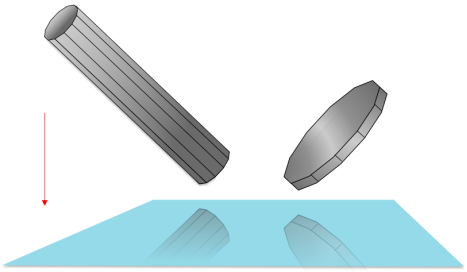

Figure 4.1: Examples of eccentric objects

is a high probability for an eccentric object being dropped on a flat surface to come to rest in a small range of stable orientations. Broadly, eccentricity is the degree to which a part deviates from being uniformly wide; generally, eccentric objects are long and thin or wide and flat.

Figure 4.1 illustrates two 3D objects in the presence of gravity on top of a flat surface which have the center of mass at their centroids. The pencil-shape object is longer in one dimension, and the coin-shape object is longer in two dimensions. It is easy to see that the pencil-shaped object is more likely to rest at one of its long sides and the coin-shaped object will rest at one of its two larger sides. Considering the set of final orientations where these objects end up with high probability, they lie on a single plane (for the pencil-shaped object) or on a single line (for the coin-shaped object). This example reveals the intuition behind our definition for geometric eccentricity. Using this definition in 3D, we show that for objects similar to pencil, there is a plane for which the final orientation is, with high probability, going to end up near; for objects similar to coin there is a line which the final orientation is going to end up near as well.

A definition for the geometric eccentricity of a planar part based on the aspect ratio of a distinguished bounding box was already shown to lead to a remarkable upper bound on the number push or squeeze actions required to orient a part with a parallel jaw gripper [133]. Here we propose a much more general notion of geometry eccentricity that not only applies in any dimension but also distinguishes between different types of object eccentricity. Using the presented definition of eccentricity, we study the problem of pose statistics for eccentric objects being dropped on a flat horizontal surface where the only force acting on the object is gravity.

Fatness is a shape-related notion that has led to improved bounds or better solutions to many problems in the field of computational geometry. There are many different definitions for fatness. (A fairly recent survey of the re-

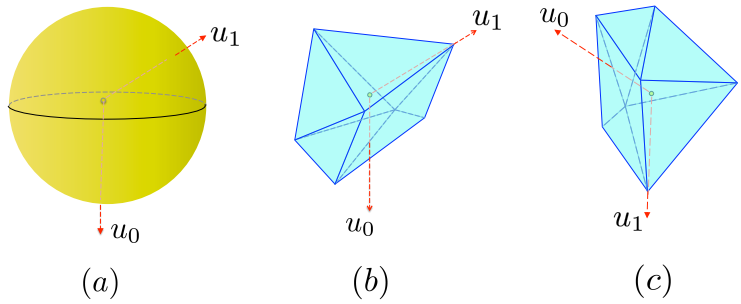

Figure 4.2: (a) Sphere of orientations in 3D and two directions u_0, u_1, (b) A convex 3D object initially at the direction u_0, (c) The same object at the direction u_1.

sults can be found in [72].) Intuitively, a fat object is an object in two or more dimensions, whose lengths in the different dimensions are similar. We note that eccentricity can be regarded as the opposite (or lack) of fatness.

In Section 4.1, we introduce the basic notion of geometric object, the quasistatic model and geometric eccentricity of the object; and in Section 4.2, we will show how eccentricity relates to the object's final pose distribution and compute bounds on the distribution for eccentric objects. Finally, we conclude in Section 4.3.

4.1 Preliminaries

We assume that the object's center of mass is given and assume without loss of generality that it lies at the origin. Since the surface onto which the object is dropped is flat, the computations can be equivalently performed on the convex hull of the object. Therefore, we assume without loss of generality that the object is convex.

4.1.1 Basic notation and definitions

We first establish some basic notation and definitions relating to the geometric properties of the convex object. We use S^{d-1} to refer to the set of *directions* in d-dimensional space, i.e. $S^{d-1} = \{u \in \mathbb{R}^d | \|u\| = 1\}$. Figuer 4.2$(a)$ illustrates orientations in S^2 and two given directionsu_0 and u_1.

We now consider a convex (closed) object $P \subseteq \mathbb{R}^d$ with center-of-mass at the origin O (where $O \in P$). The directions in S^{d-1} are attached to the initial pose of the object. We model the orientation of the object at any time as a

vector $u \in S^{d-1}$; given the object in some orientation above a flat horizontal surface, this vector is the direction of gravity, i.e. it is the unit vertical vector from the origin toward to the surface. We refer to the surface as the *floor*. Figuer 4.2(a) illustrates orientations in S^2 and two given directions u_0 and u_1. Figure 4.2(b) shows an example of an object in three dimensional, initially at the orientation u_0; in Figure 4.2(c) the object is at the orientation u_1.

We can then define the *radius function* of P as the distance from the center-of-mass to the floor of an object in a given orientation u (which is touching the floor).

Definition 4.1.1 (Radius Function)**.** *Given a convex bounded set $P \subset \mathbb{R}^d$ containing the origin, we define the* radius function*, $r_P : S^{d-1} \to \mathbb{R}$ as*

$$r_P(u) = \max_{x \in P}(x \cdot u)$$

Finally, we define the contact set *where an object in a given orientation u touches the floor.*

$$C_P(u) = \{x \in P | x \cdot u = r_P(u)\}.$$

We also note that $C_P(u)$ is convex and closed (as the intersection of two closed, convex sets, P and the plane representing the floor); thus there must be a *unique* point of $C_P(u)$ which is closest to the center-of-mass. This allows the following definition, which we refer to as the *pivot point* because (as we will see) under the quasistatic motion model the object pivots about this point if in orientation u.

Definition 4.1.2 (Pivot Point)**.** *Given object P containing O, and orientation u, $c_P(u) = \min_{x \in C_P(u)} \|x\|$.*

Definition 4.1.3 (Probability of a Set of Poses)**.** *Let Θ be a set of orientations in S^{d-1}. We define ρ as the probability that a uniformly randomly chosen unit vector is in Θ. Mathematically, $\rho(\Theta) = volume(\Theta)/volume(S^{d-1})$, where volume refers to the $(d-1)$-dimensional volume. (This resembles a generalized notion of* solid angle *in geometry.)*

4.1.2 Quasistatic model

We assume that the only force acting on the object is gravity and for simplicity we do not consider dynamics. When an object is dropped onto a flat horizontal surface, it translates downward until it contacts the surface. If the center-of-mass is not directly over a point where the object contacts the surface then it will rotate so the center-of-mass descends as quickly as possible. The same quasistatic model has been considered in some previuse work such as [140]. We refer to such an orientation as an *unstable* orientation. If the

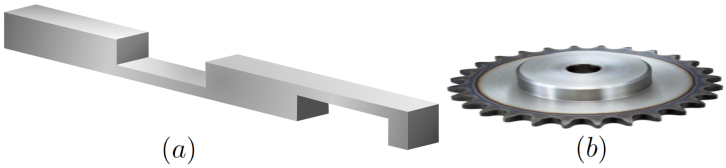

Figure 4.3: Examples of (a) an 8-(1)-eccentric object and (b) an 8-(2)-eccentric object.

center-of-mass is directly over a contact point then we have an *equilibrium* orientation. These equilibria correspond to the local mimima and maxima of the radius function. If no (local) rotation of the part will lead to a descent of the center-of-mass then the equilibrium is *stable*.

4.1.3 Geometric eccentricity

The definition of eccentricity that we propose applies to d-dimensional objects and captures the property that an object is a considerable factor, say k, bigger in b of its dimensions than in any of the remaining $d - b$ dimensions. It generalizes an earlier strictly 2D notion based on the aspect ratio of a bounding box [133] and allows us to distinguish between 3D objects similar to the pencil and similar to the coin in Figure 4.1.

Definition 4.1.4 ((b)-Eccentricity)**.** *Let $P \subset \mathbb{R}^d$ be a bounded convex set with its center of mass at the origin O. For any $1 \leq b < d$, the set P is said to be k-(b)-*eccentric *for some $k \geq 1$ if there exists a scaled and rotated copy of P such that*

- *the projection of P onto the subspace spanned by b of its orthogonal dimensions contains the b-dimensional sphere of radius k centered at O, and*
- *the projection of P onto the subspace spanned by the remaining $d - b$ orthogonal dimensions is contained in the $(d - b)$-dimensional sphere of unit radius centered at O.*

For brevity we will sometimes simply refer to an object that is (b)-eccentric for some b as an eccentric object.

Although the definition of eccentricity applies to parts in any dimension, our focus in this work is on three-dimensional (and to a lesser extent on two-dimensional) objects. Three-dimensional eccentric objects can be (1)-eccentric or (2)-eccentric: Figure 4.3 (a) and (b) show two 3D objects whose convex hulls are 8-(1)-eccentric and 8-(2)-eccentric respectively. We observe that 3D objects can be k-(1)-eccentric and k'-(2)-eccentric at the same time, for not

necessarily equal k and k'. Two-dimensional objects can only be (1)-eccentric. As a result, we will often refer to a k-(1)-eccentric 2D object as a k-eccentric object. We note that a two-dimensional object that is k-eccentric according to the definition in [133] is $k+1$-eccentric according to our definition.

4.2 Pose and probabilities

In this section, we establish lower bounds on the probabilities of a 3D object ending up in a bounded range of final poses when the initial pose is chosen uniformly at random. We will first introduce three types of orientations, then prove properties for each of these three types, and then use these properties to deduce bounds that depend on the eccentricity parameter k.

In the remainder of this work we assume without loss of generality that

- a 3D k-(1)-eccentric object is oriented and scaled such that its projection onto the x-axis contains the interval $[-k, k]$ and its projection onto the (y, z)-plane is contained in a unit disc centered at O, and
- a 3D k-(2)-eccentric object is oriented and scaled such that its projection onto the (x, z)-plane contains a disc of radius k centered at O and its projection onto the y-axis is contained in the interval $[-1, 1]$, and
- a 2D k-eccentric object is oriented and scaled such that its projection onto the x-axis contains the interval $[-k, k]$ and its projection onto the y-axis is contained in the interval $[-1, 1]$.

4.2.1 Types of poses

We focus on 3D objects with eccentricity $k > 2\sqrt{2}$. It is not surprising that we do not get results for parts with low eccentricity as such parts can have stable poses scattered all over the sphere of directions.

We first decompose S^1.

Definition 4.2.1 (Orientation Types in S^1)**.** *For a given $k > 2\sqrt{2}$, let D be the disc in $\mathbb{R}^2$ with diagonal OA_1, where $A_1 = (k, -1)$ (see Figure 4.4). The boundary of the disc D has two intersection points with $y = 1$ since $k > 2\sqrt{2}$. Let A_2 and A_3 be the intersection points with the larger and smaller x-coordinates respectively. It follows that $A_2 = ((k + \sqrt{k^2 - 8})/2, 1)$ and $A_3 = ((k - \sqrt{k^2 - 8})/2, 1)$. In addition, let $W = (0, 1)$ and V be the intersection of D with $y = 0$; note that $V = (k, 0)$. The orientations in $[0, \pi/2]$, or, in other words, the directions between OV and OW, are divided into three sectors by OA_2 and OA_3. Consider the orientations in S^1 (which correspond to rays emanating from the origin):*

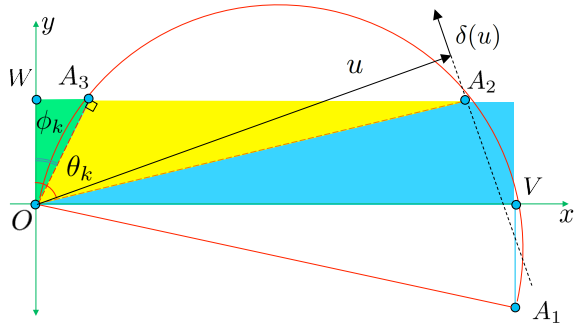

Figure 4.4: Types of orientations

- *The directions between OV and OA_2 and their images after mirroring in x-axis, y-axis, and O are* type-1 *orientations.*
- *The directions between OA_2 and OA_3 and their images after mirroring in x-axis, y-axis, and O are* type-2 *orientations.*
- *The directions between OA_3 and OW and their images after mirroring in x-axis, y-axis, and O are* type-3 *orientations.*

The type-1, type-2, and type-3 orientations will be displayed as *blue*, *yellow*, and *green* (respectively) in our figures.

The decomposition of S^1 induces separate decompositions of S^2 into orientation types associated with (1)-eccentric and (2)-eccentric objects.

Definition 4.2.2 (Orientation Types in S^2)**.** *Let $k > 2\sqrt{2}$.*

- *The orientation types in S^2 associated with (1)-eccentric parts are obtained by rotating the decomposition of S^1 about the x-axis.*
- *The orientation types in S^2 associated with (2)-eccentric parts are obtained by rotating the decomposition of S^1 about the y-axis.*

Figure 4.5 shows both decompositions of S^2.

The following property connects the eccentricity of 3D objects to the eccentricity of their 2D projections.

Property 4.2.1. *Let P be a 3D object:*

- *The projection of P onto any plane containing the x-axis is a 2D k-eccentric object if P is k-(1)-eccentric.*

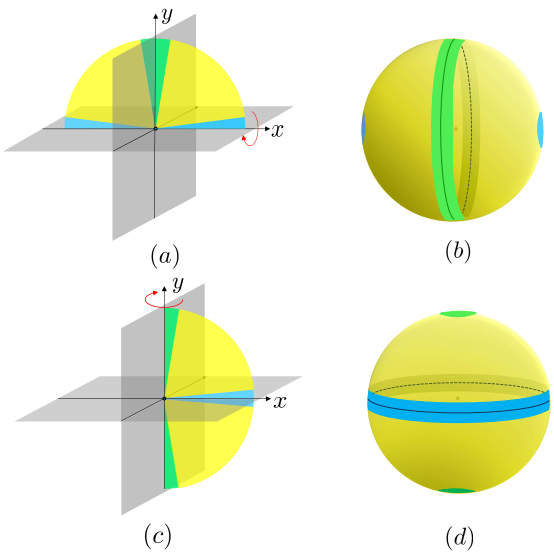

Figure 4.5: (a) The three types of orientations in a xy^+ plane for a k-eccentric 2D object. **(b)** Representation of *all* orientations for a k-(1)-eccentric 3D object. **(c)** The three types of orientations in a x^+y plane for a k-eccentric 2D object. **(d)** Representation of *all* orientations for a k-(2)-eccentric 3D object. Any k-(1)-eccentric or k-(2)-eccentric object that is initially in a type-2 or type-3 orientation will end up at a type-3 orientation.

- *The projection of P onto any plane containing the y-axis is a 2D k-eccentric object if P is k-(2)-eccentric.*

4.2.2 Properties of poses

In this subsection we will establish that a 3D object that is initially in a type-2 or type-3 pose must end up in a type-3 pose.

Lemma 4.2.2. *Type-2 poses are unstable.*

Proof. Let $P \subset \mathbb{R}^3$ be k-(1)-eccentric or k-(2)-eccentric and assume for contradiction that it has a type-2 stable pose θ in S^2. Let Z be the tangent plane to P perpendicular to θ. Since θ is stable, the point of contact between Z and P must lie on θ. Now we rotate P, along with θ and Z, around the x-axis if P is k-(1)-eccentric and around the y-axis if P is k-(2)-eccentric to get (without changing the situation) θ to lie in the (x, y)-plane. Let Q be the projection of P onto the (x, y)-plane. By Property 4.2.1, Q is a k-eccentric planar object.

Note that within the (x, y)-plane the direction θ now corresponds to a type-2 pose in S^1. If θ cannot be stable for Q then it cannot be stable for P since the supporting line of Q perpendicular to θ equals the intersection of Z with the (x, y)-plane. Therefore, it is enough to show that θ cannot correspond to a stable pose for Q. Note that a tangent line of Q perpendicular to any stable orientation, has a contact point at which the normal vector passes through the center of mass. In addition, since Q is a k-eccentric planar object, it has a point on $x = k$ in the interval $[-1, 1]$ for y. According to the definition of planar type-2 orientations, θ lies between OA_2 and OA_3. For any stable orientation, in this range (between OA_2 and OA_3) the perpendicular tangent line does not intersect the line $x = k$ in the interval $[-1, 1]$ for y, which is in contradiction with the assumption that Q is convex, thus proving the lemma. □

Lemma 4.2.3. *A k-(1)-eccentric or k-(2)-eccentric object $P \subset \mathbb{R}^3$ that is initially at a type-2 or type-3 pose ends up at a type-3 pose.*

Proof. We define the *long axes* as the x-axis in the k-(1)-eccentric case and the (x, z)-plane in the k-(2)-eccentric case. For any orientation u, we define $\alpha(u)$ to be the minimum angle between u and any orientation v contained in the long axes. We note that trivially, $\alpha(u)$ is smaller for any type-1 orientation than for any type-2 or type-3 orientation (i.e. if u_1 is type-1 and u_2 is not type-1, then $\alpha(u_1) < \alpha(u_2)$).

What we now show is that as the part rotates, $\alpha(u)$ cannot decrease; this means that if it starts in a type-2 or type-3 orientation, it cannot enter the region of type-1 orientations since this would mean decreasing $\alpha(u)$. Thus, it must settle on a non-type-1 stable pose; however, since there are no type-2 stable poses (by Lemma 4.2.2), this means that the final pose must be type-3.

We base our proof on the notation given in Figure 4.4. Suppose that the current orientation u is type-2. Without loss of generality, we can assume that u lies on the (x, y)-plane. Note that for a k-(1)-eccentric object it can be rotated around x-axes and for a k-(2)-eccentric object the object can be rotated around y-axes. Then, u corresponds to the yellow range in Figure 4.4. We note that in this case, the pivot point $c_P(u)$ (see Definition 4.1.2) must be a type-1 orientation (not necessarily in the (x, y)-plane) and therefore closer (in angle) to V than u is to V; this is because the $c_P(u)$ must be between $y = -1$ and $y = 1$. Thus, if it weren't closer to V then $c_P(u) \cdot u$ is less than the distance of any point on the far right side of $x = k$ projected onto the line containing u; but due to the object eccentricity there must be some point v there, and in that case $v \cdot u > c_P(u) \cdot u$. But this contradicts the definition of $c_P(u)$, thus proving that

$$\angle c_P(u)OV < \angle uOV.$$

We now consider the direction which the object is rotating in when it is in orientation u; this can be expressed by a unit vector $\delta(u)$ which is perpendic-

ular to u, i.e. $u \cdot \delta(u) = 0$. Note that $\delta(u)$ is in the plane passing through u, $c_P(u)$ and the origin. Suppose we are rotating in such a way as to get closer to $c_P(u)$, i.e. if $\delta(u) \cdot (c_P(u) - u) > 0$; this will cause the radius to increase (i.e. the center of mass to rise). This is impossible, so $\delta(u) \cdot (c_P(u) - u) \leq 0$. But since $c_P(u)$ is closer in angle to V than u is,

$$\delta(u) \cdot (c_P(u) - u) \leq 0 \implies \delta(u) \cdot (V - u) \leq 0.$$

Thus, the angle between the current pose u and V (the long axis in Figure 4.4) is never increasing as long as u is type-2. Thus, starting from a type-2 or type-3 orientation cannot lead to a type-1 final orientation, completing the proof. □

Lemma 4.2.2 shows that type-2 orientations are unstable. The remaining orientations are *potentially stable* orientations. The stable orientations in type-1 have radii larger than $|OA_2|$ the stable orientations in type-3 have radii smaller than $|OA_3|$.

Lemmas 4.2.2 and 4.2.3 and the trivial fact that a part cannot settle in an unstable orientation reveal that objects that are initially in a type-2 or type-3 orientation will end up in type-3 orientation and objects that are initially in a type-1 orientation will end up in a type-1 or type-3 orientation. In other words, objects in yellow or green poses in Fiure. 4.5 will end up in a green final pose and objects in a blue pose will end up in blue or green final pose. The following theorem summarizes the former (non-trivial) part of this conclusion in terms of probabilities.

Theorem 4.2.4. *Let $P \subset \mathbb{R}^3$ be a k-(1)-eccentric or k-(2)-eccentric object ($k > 2\sqrt{2}$) in a uniformly random initial orientation. The probability of P ending up at type-3 orientations is at least $\rho(\Theta)$ where Θ is the union of the set of type-2 and type-3 orientations.*

We observe that the final orientations lie in the proximity of a plane, the (y, z)-plane if P is k-(1)-eccentric and the (x, z)-plane if P is k-(2)-eccentric, and the proximity of a line, the x-axis if P is k-(1)-eccentric the y-axis if P is k-(2)-eccentric. Theorem 4.2.4 gives a lower bound on the probability for (or bias towards) the plane if P is k-(1)-eccentric and the line if P is k-(2)-eccentric. In the next subsection we will see that lower bounds as well as the sizes of the proximities depends on k.

4.2.3 Computing the probabilities

In this subsection we will obtain an upper bound on the size of the set of type-3 orientations in which the part will settle with a probability that equals at least the fraction of S^2 covered by type-2 and type-3 orientations. We will also

derive an upper bound on the size of the set of type-1 and type-3 orientations in which the object is guaranteed to settle. Although similar bounds for 2D objects are easy to obtain we focus solely on 3D objects. We define $\Phi_k^{(i)}$, $\Theta_k^{(i)}$, and $\Psi_k^{(i)}$ (with $i = 1, 2$) to be the sets of all type-3, union of type-2 *and* type-3, and type-1 orientations in S^2 for a k-(i)-eccentric object, respectively.

We recall Definition 4.2.1 and let $\theta_k = \measuredangle A_2OW$ and $\phi_k = \measuredangle A_3OW$. Note that θ_k and ϕ_k are the angles in S^1 relative to the y-axis that mark the boundaries between type-1 and type-2 orientations and between type-2 and type-3 orientations respectively. By Definition 4.2.2 these angles also determine the boundaries between the corresponding types of orientations in S^2. (See Figure. 4.4.) From Definition 4.2.1 it follows that

$$\theta_k = \arctan\left(\frac{k + \sqrt{k^2 - 8}}{2}\right) \tag{4.1}$$

$$\phi_k = \arctan\left(\frac{k - \sqrt{k^2 - 8}}{2}\right) \tag{4.2}$$

Using the well-known fact that the surface area of a segment or cap of height h of the unit sphere equals $2\pi h$ and observing that $\Phi_k^{(i)}$, $\Theta_k^{(i)}$, and $\Psi_k^{(i)}$ can all be regarded as unions of two segments or caps we obtain

$$\begin{aligned}
\rho(\Theta_k^{(1)}) &= \frac{2}{4\pi}(2\pi \sin\theta_k) = \sin\theta_k \\
\rho(\Phi_k^{(1)}) &= \frac{2}{4\pi}(2\pi \sin\phi_k) = \sin\phi_k \\
\rho(\Psi_k^{(1)}) &= \frac{2}{4\pi}(2\pi(1 - \sin\theta_k)) = 1 - \sin\theta_k,
\end{aligned} \tag{4.3}$$

and we also have

$$\begin{aligned}
\rho(\Theta_k^{(2)}) &= \frac{2}{4\pi}(2\pi(1 - \cos\theta_k)) = 1 - \cos\theta_k \\
\rho(\Phi_k^{(2)}) &= \frac{2}{4\pi}(2\pi(1 - \cos\phi_k)) = 1 - \cos\phi_k \\
\rho(\Psi_k^{(2)}) &= \frac{2}{4\pi}(2\pi \cos\theta_k)) = \cos\theta_k.
\end{aligned} \tag{4.4}$$

We can now use these equations, Theorem 4.2.4, and the observation that no part can settle in a type-2 pose to formulate our main result.

Theorem 4.2.5. *Let $P \subset \mathbb{R}^3$ be a k-(1)-eccentric or k-(2)-eccentric object ($k > 2\sqrt{2}$) in a uniformly random initial orientation, and let $\theta_k = \arctan\left((k + \sqrt{k^2 - 8})/2\right)$ and $\phi_k = \arctan\left((k - \sqrt{k^2 - 8})/2\right)$. If P is k-(1)-eccentric then it settles*

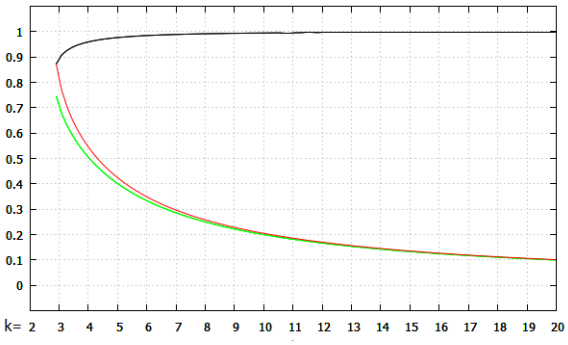

Figure 4.6: The graph of $\rho(\Theta_k^{(1)})$, $\rho(\Phi_k^{(1)})$, and $\rho(\Phi_k^{(1)} \cup \Psi_k^{(1)})$ illustrated in black, green, and red, respectively for a k-(1)-eccentric object.

- *with probability at least* $\sin\theta_k$ *in a spherical segment of orientations symmetrically surrounding a plane that covers a fraction* $\sin\phi_k$ *of* S^2*,*
- *and in one of two antipodal spherical caps of orientations surrounding a line that jointly cover a fraction* $1 - \sin\theta_k$ *of* S^2 *otherwise.*

If P is k-(2)-eccentric then it settles

- *with probability at least* $1 - \cos\theta_k$ *in one of two antipodal spherical caps of orientations surrounding a line that jointly cover a fraction* $1 - \cos\phi_k$ *of* S^2
- *and in one spherical segment of orientations symmetrically surrounding a plane that covers a fraction* $\cos\theta_k$ *of* S^2 *otherwise.*

The theorem shows that the sizes of the caps and segment in which P is guaranteed to settle shrink if k increases, so the uncertainty about the final pose decreases. Moreover, it shows that the bias towards the segment (or plane) if P is k-(1)-eccentric and towards the caps (or line) if P is k-(2)-eccentric increases if k increases (at the cost of the caps and segment respectively). Another way of stating the theorem is as follows; for k-(1)-eccentric objects, with probability $sin\theta_k$, the final orientation has at most ϕ_k distance from a plane and for k-(2)-eccentric objects, with probability $1 - cos\theta_k$, the final orientation has at most $\pi/2 - \theta_k$ distance from a line. Figure 4.6 and 4.7 provide insight into how the sizes of the sets and the probabilities vary as a function of k.

Figure 4.6 shows that a 8-(1)-eccentric part always settles in a region containing less than 26% of the orientations and with a probability that is close to 99% in a region containing less than 25% of the orientations. Figure 4.7

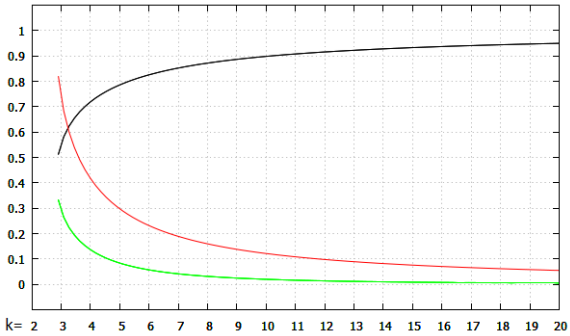

Figure 4.7: The graph of $\rho(\Theta_k^{(2)})$ and $\rho(\Phi_k^{(2)})$, and $\rho(\Phi_k^{(2)} \cup \Psi_k^{(2)})$ illustrated in black, green, and red, respectively for a k-(2)-eccentric object.

shows that a 8-(2)-eccentric part always settles in a region of less than 15% of the orientations and with a probability that is close to 89% in less than 4% of the orientations. It can be observed that for larger k, there is a smaller range of poses that the object always can settle in and there is a higher probability for the object to settle in again smaller range of poses.

4.3 Conclusions

In this chapter, we studied the pose statistics problem for a family of 3D objects with initial pose, uniformly at random. We assumed that the object falls onto a flat surface in the presence of gravity under the quasi-static conditions. We proposed a novel type of geometric eccentricity for d-dimensional objects for which the final distribution of 3D objects have a substantial bias towards a small range of poses when they come to rest. This result can be applied to part feeding tasks to obtain a faster design for reducing the object pose uncertainty.

According to our proposed notion of eccentricity, for a given k, we have two types of eccentricity in 3D: k-(1)-eccentric objects are k times bigger in one dimension than the other two dimensions while k-(2)-eccentric objects are k times bigger in two dimensions than the other dimension. We showed that for both types of eccentric objects there is a small cluster of poses at which the object always ends up. We showed that there is a high probability of ending up at a pose which is close to a specific plane for k-(1)-eccentric objects, and close to a specific line for k-(2)-eccentric objects. In addition, we showed that for larger k, there is smaller range of poses (which are close together) at which

the k-(1)-eccentric and k-(2)-eccentric objects end up with higher probability.

We defined a notion of *geometric eccentricity* for d-dimensional objects, which takes into account both its shape and its center-of-mass. We show that under quasi-static conditions, the pose into which eccentric objects settle will be with high probability in a cluster of poses which are very close together. Furthermore, the probability of ending up in this range of poses increases, and the size of the range decreases, as the object gets more eccentric. Our approach can be applied for planar parts as well as parts with shape variation under the proposed model in Chapter 1.

Our computation has been simplified by the quasi-static assumption. However, it is has been shown that under a more realistic model a better result (higher probability to end up in a smaller range of poses) is expected [71, 134]. It implies that our bounds hold in practice. To rigorously show this, in future work, the model incorporating the dynamics can be explored.

Our approach to analyzing the distribution of the final orientations works based on the assumption that the object is k-(1)-eccentric or k-(2)-eccentric. Therefore, finding the largest k such that the object is k-(1)-eccentric/k-(2)-eccentric remains is an interesting area for future research.

PART III

GEOMETRIC PROBING

Chapter 5

Proximity probing algorithms for metrology

"The real voyage of discovery consists not in seeking new landscapes, but in having new eyes."

- Marcel Proust

METROLOGY is the theoretical and practical study of measurement. One of the important aspects of metrology which has applications in manipulation and robotics is the process of learning the shape and orientation of an unknown object using a simple measurement device called *probe*. When the measurements relate to geometric properties such as a distance from an object or the diameter of the object, then the approach is called *geometric probing*. Geometric probing is a useful technique in any situation where a complex property is being studied with a sensing device which is not powerful enough to determine the desired information without multiple uses; such a situation often arises when cost, robustness, or feasibility constraints prevent more complex sensors from being used, and occurs in fields such as robotics.

There are different types of geometric probing which have been reviewed in Chapter 1. In this chapter, we study a type of proximity probing related to the class of geometric probing in literature. Given a point, it returns the distance to the boundary of the nearest object. See Figure 5.1. This type of probe is particularly interesting when one needs to sense the object without

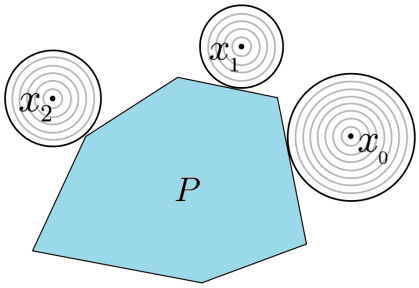

Figure 5.1: An unknown polygon P with proximity probes at x_0, x_1, x_2

providing any directional information.

When there is an unknown convex polygon P in the plane, the goal is to minimize the number of probe measurements needed to exactly determine the shape and location of P. We refer to this as *Problem 1* and present an algorithm to solve it, as illustrated in Figure 5.3. In many cases, the object is not known while a set of finite possible objects is given. This paper also includes an algorithm and analysis of *Problem 2*: identifying a convex polygon from a known set.

The remainder of the chapter is organized as follows. In Section 5.1, we introduce Problem 1 and the definitions necessary for the algorithms. In Section 5.2, we present our algorithm for Problem 1 and analyze its complexity per probe; we also present a complete example of the algorithm for a simple polygon P. In Section 5.3 we present an upper bound on the number of probes needed by the algorithm. Section 5.4 deals with Problem 2, presenting and analyzing an algorithm aiming to minimize the number of probes necessary to identify P from the set. Finally, in Section 5.5, we summarize our results and discuss future work.

5.1 Preliminaries

We assume that all points and objects lie in the plane and that all positioning and measurements are exact.

For any two points or closed sets of points a, b, $\mathrm{dist}(a, b)$ denotes the Euclidean distance between a and b; for a closed subset S of the plane, $\partial(S)$ denotes its boundary, $\mathrm{Int}(S)$ denotes its interior, $\overline{S}$ denotes the closure of its complement (so both S and $\overline{S}$ contain $\partial(S)$), and $\mathrm{Conv}(S)$ denotes its convex hull. We also define *zero-disk* to mean a disk containing only its center.

In addition, for any disk of positive radius C and point z on its boundary,

we define $L(C, z)$ to be the line tangent to C at z. We also define $H(C, z)$ to be the half-plane bordered by $L(C, z)$ which contains C.

5.1.1 Problem 1: Formulation and definitions

Let P be an unknown convex polygon with n vertices and edges contained in a known disk D, and let the *probing function* f_P be defined over the plane as

$$f_P(x) = \begin{cases} \text{dist}(x, P) & : x \notin \text{Int}(P) \\ -1 & : x \in \text{Int}(P) \end{cases}$$

The probing algorithm is not explicitly given this function, but is allowed to call it as many times as necessary to find P exactly; the goal of this work is to find an algorithm which minimizes the upper bound of probes necessary (and allows the next probe to be efficiently computed at each step). The points x for which it calls the function f_P are the *probes*, and the disks of radius $f_P(x)$ centered at these points are the *probe disks*, abbreviated as *p-disks* (by convention, if $f_P(x) = -1$ then no disk is produced). Every p-disk is by definition incident to P at exactly one point.

5.1.2 Condensed probe disks

In order to properly describe how our algorithm uses these p-disks to extract the shape of the object P, we first must discuss an especially important configuration of p-disks in which we can simplify the situation. In the case where one p-disk contains another, we can conceptually replace both of them with a zero-disk and an associated line and half-plane with an operation we refer to as *condensation*. We refer to these zero-disks and their associated lines and half-planes as *condensed probe disks* or *cp-disks*. Our algorithm automatically performs this operation whenever it can, and thus explicitly uses cp-disks in its main steps.

Suppose we have two (distinct) p-disks C_a, C_b such that $C_a \subset C_b$. Since they both must be incident to P at exactly one point, they must be incident to P at the same point (otherwise it is impossible for one to contain the other); this point will by definition be the only point in $\partial(C_a) \cap \partial(C_b)$, which we call $p_{a,b}$. Furthermore, P must be interior disjoint with the half-plane $H(C_b, p_{a,b})$ since P is convex, $p_{a,b} \in C_b, P$, and C_b is not a zero-disk (because $\emptyset \neq C_a \subset C_b$). The condense operation on C_a, C_b is thus defined as the function which outputs $p_{a,b}$ as a zero-disk and associates with it the half-plane $H(C_b, p_{a,b})$. Furthermore, any p-disks which neither contain nor are contained by other p-disks (and so cannot be used by the condense operation) are also considered to be cp-disks. Note that cp-disks, like p-disks,

must have exactly one intersection point with P (since cp-disks are either p-disks or zero-disks produced by the condense operation). If C^* is a cp-disk produced by the condense operation (and hence C^* is a point), we let $H(C^*)$ be its associated half-plane and $L(C^*)$ be the line bordering $H(C^*)$.

Remark: It is possible for a p-disk C_a to be contained in several other p-disks, none of which are contained in each other; however, this can only happen when C_a is a zero-disk and also at a vertex of P. In these cases, C_a will be condensed with every disk containing it to produce multiple cp-disks.

5.1.3 Shadow sets

In order to effectively choose the next point to probe from, at each step our algorithm computes the *infeasible* region, a region of the plane which cannot contain any point of the polygon P. The infeasible region increases at each step until all the edges and vertices of P are determined. See Figure 5.3. This is done primarily by using the convexity of the polygon, along with the fact that it must be incident to (but not present in the interior of) each cp-disk. Thus, if we select a point such that there is a cp-disk such that every line from the point to the cp-disk passes through the interior of another cp-disk, this point cannot be part of P (since in order to contain this point and be incident to both cp-disks, P would have to be nonconvex). We refer to the sets of these points as *shadow sets* because if we treat any cp-disk as a light source, they would be the shadows cast by the other p-disks. The infeasible region is then the union of these shadow sets, with certain gaps filled in as well (as we will describe later).

Suppose we have two cp-disks C_i^*, C_j^*; for both we define a counterclockwise direction on their boundary. We define the lines $L_{i,j}$ and $L'_{i,j}$ to be the lines tangent to both C_i^* and C_j^* such that

- for both lines, C_i^* and C_j^* lie on the same side
- $L_{i,j}$ is given a direction coinciding with the counterclockwise direction imposed on the two cp-disks, while $L'_{i,j}$ is given a direction opposing the counterclockwise direction
- Both lines, intersect C_i^* before C_j^*, by considering the directions of them.

Note that $L_{i,j}$ is the same line as $L'_{j,i}$ but with the opposite direction imposed on it.

We now define the rays $l_{i,j}, l'_{i,j}$ to be the rays respectively lying on $L_{i,j}, L'_{i,j}$ with their sources at the respective points of tangency with C_j^*. For a ray l, we define $H_{right}(l)$ to be the quarter plane lying directly to the right of the ray, and $H_{left}(l)$ is analogously defined.

The shadow set cast by C_j^* on C_i^* is then

$$S_i(j) = C_j^* \cup (H_{left}(l_{i,j}) \cap H_{right}(l'_{i,j}))$$

This set cannot contain any point of $\mathrm{Int}(P)$, since P cannot have any point in $\mathrm{Int}(C_j^*)$, must be incident to C_i^*, and is convex; similarly, P cannot contain any point of $\mathrm{Int}(S_i(j))$. Figure 5.2 illustrates shadow sets for three consecutive cp-disks.

Note that the boundary of C_j^* is partly on the boundary of $S_i(j)$ and partly in its interior; since P cannot contain any point of $\mathrm{Int}(S_i(j))$, its point of intersection with C_j^* must be on the part of $\partial(C_j^*)$ which is also on $\partial(S_i(j))$. We call this the *feasible arc* C_i^* imposes on C_j^* and denote it $\zeta_i(j)$.

5.1.4 Clockwise ordering of cp-disks

One of the main advantages of our algorithm over less intricate alternatives is that it provides for a constant-time computation for each probe; in order to achieve this, it must find a way to avoid explicitly computing all the shadow sets produced with each new cp-disk, as this would lead to geometric growth of the necessary computation as the algorithm continues. We do this by specifying a circular ordering on the cp-disks and then showing that the feasible region can be computed using only the shadow sets cast by neighbors in this ordering. Thus, in order to update our representation of the infeasible region after the insertion or alteration of any cp-disk in the ordering, only the shadow sets involving the inserted or altered cp-disk and its neighbors must be computed.

For any cp-disk C^*, let $p(C^*)$ be its intersection point with P. We note that by imposing a clockwise direction on the boundary of P, we can impose a clockwise order on the set of $p(C^*)$ for all cp-disks C^* (it is possible for two cp-disks to have the same contact point on P; but this can only happen on vertices of P). This then imposes a clockwise (cyclic) ordering on the set of cp-disks, where if multiple cp-disks happen to have the same contact point with P, they can be ordered by the lines tangent to them at the common contact point (a zero-disk C_{zero}^* produced by the condense operation is considered to have the line $L(C_{zero}^*)$ as its tangent; a zero-radius cp-disk not produced by the condense operation cannot share a contact point with another p-disk or cp-disk since it would be contained by the other disk and hence not be a cp-disk).

From now on we will attach indices to the cp-disks indicating their order. Specifically, we will let S_{disks} be the ordered set of cp-disks, and implicitly label the disks in S_{disks} as $C_1^*, C_2^*, ..., C_\alpha^*$. Since the ordering of cp-disks is cyclic,

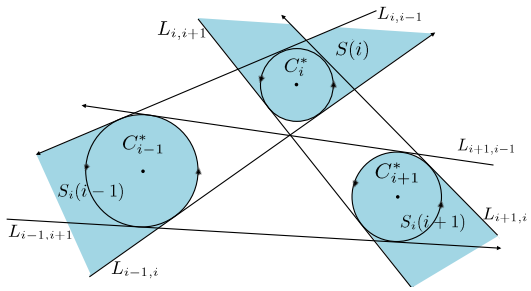

Figure 5.2: An example of three consecutive cp-disks, which shows the shadow sets of C_{i-1}^*, C_{i+1}^* with respect to C_i^* and the neighbor-shadow set of C_i^*.

we assume that additions and subtractions on indices are performed modulo the number of disks.

Remark: It should be noted that in general, given an arbitrarily probed set of cp-disks, prior knowledge of P might be necessary to deduce their exact ordering by the above criteria, and thus the ordering cannot be used by the algorithm. However, we will show that our algorithm chooses probes in such a way that this labeling can always be determined exactly without any prior knowledge of P.

5.1.5 The infeasible region

We first define the set $S_{i-1}(i) \cup S_{i+1}(i)$ to be the *neighbor-shadow set* of C_i^* (abbreviated as *ns-set*), denoted as $S(i)$ for convenience (Figure 5.2). Similarly, we define the set $\zeta_{i-1}(i) \cap \zeta_{i+1}(i)$ to be the *neighbor-feasible arc* of C_i^*, abbreviated as *nf-arc*; we denote it as $\zeta(i)$ for convenience. The nf-arc essentially denotes, for any cp-disk, the arc of the disk where its intersection with P can occur (so that it is consistent with the neighboring cp-disks).

For cp-disks produced by the condense operation, we instead use $S(i)$ to refer to the half-plane $H(C_i^*)$. Note that since no cp-disk can be contained in $\text{Int}(H(C_i^*))$, $H(C_i^*)$ is a superset of $S_{i-1}(i) \cup S_{i+1}(i)$ for these cp-disks.

The *infeasible region* R can now be defined as

$$R = \bigcup_{i=1}^{m} S(i) \cup \overline{D}$$

Intuitively, for each C_i^*, we simply take the ns-set of C_i^*, the half-planes associated with all cp-disks generated by the condense function, and the complement of D (the disk which we were initially given as containing P). Since

R is composed of these pieces, P must be entirely contained in (the closure of) the complement of R.

We will show later that our algorithm behaves in such a way that the complement of R (the *neighbor-feasible region*) is a single connected piece; we therefore will assume it to be the case now. The boundary of R will then be split into the following two basic types of pieces, which we call *sections*:

1. arcs of the boundary of D,
2. connected subsets of the boundaries of the sets $S(i)$; we denote $\partial(S(i)) \cap \partial(R)$ as $\partial_R(S(i))$.

Note that the second type of section has two possibilities:

a) if C_i^* was not produced through the condense operation, $\partial_R(S(i))$ is naturally split into at most three pieces, namely

- the nf-arc $\zeta(i)$,
- a segment of the ray $l_{i-1}(i)$ (which we will denote $l(i)$ for convenience),
- a segment of the ray $l'_{i+1}(i)$ (which we will denote $l'(i)$ for convenience).

We note that the other two pieces of the boundary of $S(i)$, namely $l'_{i-1}(i)$ and $l_{i+1}(i)$ cannot lie on $\partial(R) = \partial(\overline{R})$ because P in that case would impose the wrong ordering of the cp-disks.

b) if C_i^* was produced through the condense operation, $\partial_R(S(i))$ is just $L(C_i^*)$

Remark: Although the infeasible set R is interior disjoint with P by definition, it is not necessarily the case that it is the full set of all infeasible points, i.e., the points which, given the p-disks, can't be contained in P.

5.1.6 Confirmation of vertices and edges, and the query set

We say a point v is *confirmed* if by considering $S_{disks} = \{C_1^*, C_2^*, ..., C_\alpha^*\}$ it can be shown that v is a vertex of P, and we say a line L is confirmed if by considering S_{disks} it can be shown that l contains an edge of P; an edge e of P is also referred to as confirmed if the line extending it is confirmed. Any vertices or edges of P which are not confirmed are called *unconfirmed*. The list of confirmed vertices is denoted by V_c and the list of confirmed edges is denoted by E_c. Our algorithm determines P by systematically confirming its edges and vertices.

Now we consider $\partial(R)$, as described above as a collection of pieces of the boundaries of the $S(i)$ and D. Since $\partial(R)$ is continuous, there will be points which lie on more than one of the specified sections. Some of these points will lie on confirmed vertices or edges of P. The ones which do not will be called the *query set* Q, from which we will always probe (except for the very first probe). Furthermore, we define the *preferred query set* Q^* to be the subset of Q which does not contain any intersection points between two p-disks.

To confirm a vertex or line, we need to count how many p-disks are incident to it; an easy way to compute this from the set of cp-disks is to count the number of cp-disks tangent to L, double-counting those produced by the condense operation (since they correspond to two p-disks). Note that this means the number of cp-disks involved is at most the number of p-disks involved.

Furthermore, note that the set of all cp-disks passing through a point or tangent to a line must be consecutive.

We can confirm a point v as a vertex of P in these cases:

- if 3 p-disks pass through v,
- if v is probed and $f_P(v) = 0$ (this implies that $v \in \partial(P)$; the fact that v was in Q, which is a necessary condition for being probed by the algorithm, means that v sits in a corner of R and thus cannot be in the middle of an edge of P, meaning it must be a vertex of P),
- if a segment of (confirmed or unconfirmed) line L on $\partial(R)$ and two p-disks touch v,
- if segments of (confirmed or unconfirmed) lines L, L' on $\partial(R)$ and one p-disk touch v.

If we confirm a vertex on a previously unconfirmed line, we can automatically confirm the line as well.

Additionally, we can confirm a line L as containing an edge of P in these cases:

- if L is tangent to three p-disks.

  The cp-disks representing these three p-disks will necessarily be consecutive in S_{disks} because they all have contact points with P on the same edge (and no other cp-disks will have contact points in the interior of this edge, since in that case L would have been confirmed earlier), and so given a cp-disk C_i^* we just need to check the three consecutive triples containing it.

- if line L is tangent to two p-disks and passes through the intersection point v of the boundaries of two other p-disks, then both L and v can be confirmed.

- if L is tangent to a p-disk and goes through the intersections of the boundaries of two different pairs of p-disks (call these points v_1, v_2), we can confirm v_1, v_2 and L.

Whenever a vertex v is confirmed, it automatically implies that probing v would return $f_P(v) = 0$; this means we can place a p-disk there *without* explicitly probing it, and perform the condense operation with any existing p-disks which happen to contain v. Since they all have the same contact point v with P, they will be consecutive in S_{disks}, and later on we will show that there cannot be more than 3 such disks for any v, so this process takes constant time.

Similarly, whenever a line L is confirmed, we always have at least one, and often more than one, cp-disks tangent to L; at each tangent point x we know that $f_P(x) = 0$ so we may place a p-disk there without actually executing the probe function, and perform the condense operation with the original tangent p-disk to create a new cp-disk. Since an edge is always confirmed if it is incident to 3 cp-disks, the number of condense operations we need to perform is at most 3 for each confirmed line; thus this process takes constant time.

Remark: Thanks to the fact that we use the condense operation when we confirm vertices and edges (without requiring new probes), the lines corresponding to these condense operations are automatically incorporated into $\partial(R)$.

5.2 Algorithm

We now present an efficient algorithm for solving the probing problem described in Section II. The general strategy is to start by probing from some specific points arbitrarily until a first edge is confirmed (e_1 in Figure 5.3). Then, the algorithm tries to confirm the next edges in clockwise order (e_2, e_3, e_4 in Figure 5.3). The algorithm maintains the circular ordered list of cp-disks, S_{disks}, sorted in clockwise order of their intersection point with P around $\partial(P)$, an algebraic representation of the infeasible region R, lists of the confirmed vertices (V_c) and edges (E_c) of P, and representations of the query set Q and preferred query set Q^*. We present it in two parts: the first dealing with how to generate the next probe given S_{disks}, R, V_c, E_c, Q, and Q^*, and the second dealing with how to update these objects given a new probe result. The algorithm terminates once (a) at least one vertex and edge have been confirmed and (b) every confirmed vertex is on two confirmed lines and every confirmed line contains two confirmed vertices.

In addition, some extra information and pointers will be stored in these lists in order to allow the algorithm to execute all the steps in constant time, most notably pointers in Q for each element which point to its neighbors (in

both S_{disks} and Q); however, we omit the exact details.

5.2.1 Algorithm for generating new probes

The algorithm for generating new probes is divided into two distinct phases (preceded by a one-probe initialization): in Phase 1, we probe arbitrarily from the preferred query set Q^* when possible; when it is not, we choose instead from Q (both Q^* and Q are by definition a subset of the boundary of R) until some edge is confirmed; in Phase 2 (once an edge is confirmed), we probe points designed to confirm the vertices and edges of P in (roughly) clockwise order.

We also add the following definitions for reference in the algorithm:

- the first edge of P to be confirmed is denoted e_1 (i.e., the edge contained by the first line confirmed),
- the edges and vertices of P in clockwise order are $e_1, v_1, e_2, v_2, ..., e_n, v_n$,
- for any edge e_i, we let L_i^* be the line containing e_i; note that it is the lines, not the edges themselves, which are directly confirmed by the algorithm,
- at any given step of the algorithm, we let t be the largest index such that $e_1, v_1, e_2, v_2, ..., e_{t-1}$ are all confirmed (we can determine t from E_c and V_c without any extra direct knowledge of P),
- l is a ray originating on some point on e_{t-1} which we know is in P (for all $t > 2$, we use v_{t-2}; otherwise we use the contact point of some p-disk with the confirmed line containing e_{t-1}) and extending e_{t-1} in the direction coinciding with the clockwise direction around the boundary of P (this direction is also determinable from E_c and S_{disks} without any extra knowledge of P),
- for any set S and ray γ, let $\rho(\gamma, S)$ be the furthest point along γ which is also in S.

At the start, X, V_c, E_c, Q, Q^* are empty and $R = \overline{D}$, so we simply probe from an arbitrary point on the boundary of D. Because $P \subset \text{Int}(D)$, this disk will have positive radius; because it is the first p-disk, it cannot be condensed and is thus also a cp-disk. In addition, it will not have any neighbors in S_{disks} since it is the only disk in S_{disks}, so its shadow set is by convention defined to be itself. Thus, R is simply the union of this disk and the complement of D, and the boundary of R will consist of an arc of this disk plus an arc of D.

Hence, by definition, Q consists of the two points of intersection between the boundaries of D and the first cp-disk.

Algorithm Steps:

1. While no line has been confirmed, at each step we check if Q^* has at least one element. If it does, we choose an arbitrary point $x \in Q^*$ and probe it; if not, we choose an arbitrary point $x \in Q$ and probe it.

2. Once a line has been confirmed, we let the edges and vertices of P, the index t, and the ray l be defined as above. We repeat the following step until both e_t and v_{t-1} are confirmed (at which point, by definition, the index t increases, and we start Phase 2 again; we terminate once v_t is confirmed on e_1).

   Let $x = \rho(l, \overline{R})$; an intuitive idea of x is that it is the furthest clockwise point on the confirmed line containing e_{t-1} which is not in the infeasible region R. We note then that since x is the furthest point on $l \subset L^*_{t-1}$, it must also be on some other object on the boundary of R; hence, either $x \in V_c$ (if x happens to be v_{t-1} and is already confirmed) or $x \in Q$.

   If $x \in Q$ then it must be both on L^*_{t-1} and some other piece of the boundary of R. In particular, it can be on any of the following pieces:

   - an nf-arc $\zeta(i)$ of some cp-disk C^*_i,
   - another confirmed line,
   - an unconfirmed line, either corresponding to the output of a condense function or incident to two (consecutive) cp-disks,
   - the boundary of D.

   We then do the following:

   (a) if $x \in V_c$, call *Next Edge*,

   (b) if $x \in Q$ and $x \notin \zeta(i)$ for all i, probe x,

   (c) if $x \in Q$ and $x \in \zeta(i)$ for some i, then it is one endpoint of the arc $\zeta(i) \cap \partial(R)$; let x' be the other endpoint. This point by definition will either be x's neighbor in Q or will be an endpoint of $\zeta(i)$, and hence is retrievable in constant time.

Remark: Although in Phase 1 we are allowed to probe any $x \in Q^*$ (or, if Q^* is empty, any $z \in Q$) at each step, if we wish to minimize the time complexity of choosing the next probe at each step, we need a retrieval method which produces a member of Q^* or Q in constant time; having either a stack

or a queue as an additional data structure for Q^* and Q are the most natural ways of achieving this.

The Next Edge Procedure

This procedure is called when e_{t-1} and v_{t-1} are both confirmed but e_t is not confirmed and the procedure aims to confirm it. Let us consider the set of cp-disks incident to v_{t-1}; they will be consecutive in S_{disks}, and will have been produced by the condense function (at the moment that v_{t-1} was confirmed). Let C_i^* be the last cp-disk among them; let $\mathrm{N}_Q(C_i^*)$ be C_i^*'s next neighbor (in the clockwise direction) in Q. We then probe $\mathrm{N}_Q(C_i^*)$ (updating the maintained information as we go so i and $\mathrm{N}_Q(C_i^*)$ can change after each probe) until the next edge is confirmed, at which point t can be updated and we return to the main loop of Phase 2. We note that $\mathrm{N}_Q(C_i^*)$ is actually the point on $L(C_i^*)$ furthest from v_{t-1}.

The Pseudocode

For the pseudocode, we introduce some extra notation and functions (and show, where necessary, that these functions can be computed efficiently). We define the sets E_c^*, V_c^* to be respectively the subset of E_c consisting of those lines which do not contain two points from V_c, and the subset of V_c consisting of those points which are not contained by two lines from E_c. Intuitively, E_c^* and V_c^* consist of the confirmed lines and vertices whose adjacent vertices and lines, respectively, have not been confirmed yet. These sets are easy to maintain with flags attached to both E_c and V_c.

For the case (c) of Phase 2, if $x \in \zeta(i)$, then we denote the other endpoint of the arc $\zeta(i) \cap \partial(R)$ as $q(x)$.

For any $x \in Q$, we note that since we can retrieve its neighbors in S_{disks} in constant time, we can determine whether it is on some nf-arc in constant time; we will treat this as a binary valued funtion $\mathrm{nf}(x)$ which is *true* when x is on some nf-arc, and $false$ otherwise.

The *RandomElement* function refers to random or arbitrary choice of some element from a set; the *Probe* function refers to the full update algorithm (described in Section IIIB), which uses and modifies all the objects in the program. Most object updates occur within the Probe function.

Note that by the time Phase 2 starts, by definition, we will have at least one member of E_c; note also that maintaining Q^* is only necessary for Phase 1.

5.2.2 Algorithm for handling a new probe

The algorithm for updating the maintained information (S_{disks}, R, E_c, V_c, Q, Q^*) is relatively simple since we usually probe from the set Q (since $Q^* \subset Q$). To update S_{disks} in this case, we merely note that each point $x \in Q$ is specifically linked to two consecutive 'neighbors' in S_{disks}.

Algorithm 3 Identifying P using proximity probes

procedure DETERMINEP(D)
  $V_c, E_c \leftarrow null$ ▷ Initialization
  $\partial(R) \leftarrow \partial(D)$
  $x \leftarrow$ RandomElement($\partial(D)$)
  run Probe(x)
  while $E_c = null$ **do** ▷ Phase 1
    if $Q^* \neq null$ **then**
      $x \leftarrow$ RandomElement(Q^*)
    else
      $x \leftarrow$ RandomElement(Q)
    run Probe(x)
  while $E_c^* \neq null$ and $V_c^* \neq null$ ▷ Phase 2
    $x \leftarrow \rho(L, R)$
    if $x \in V_c$ **then** ▷ Case a:
      run NextEdge(x) ▷ $x = v_{t-1}$
    else if nf(x) $= false$ **then** ▷ Case b:
      run Probe(x) ▷ x is not on an nf-arc
    else ▷ Case c:
      $x' \leftarrow q(x)$ ▷ x is on an nf-arc
      run Probe(x')
  return V_c ▷ Return P as a set of vertices
end procedure

procedure NEXTEDGE(x)
  while $\neg\exists e \in (E_c \backslash e_{t-1}) | x \in e$
    $x' \leftarrow \mathrm{N}_Q(C_i^*)$
    run Probe(x')
end procedure

If the new p-disk contains or is contained by one or both of the 'neighbor' cp-disks of its center, we perform the condense operation; this check trivially takes constant time since it has only two neighbors. It cannot contain or be contained by any non-neighboring cp-disks, and therefore checking whether the condense operation has to be used has constant time complexity per step.

The only case where we do not probe from Q is in Phase 2, when line L^*_{t-1} containing edge e_{t-1} meets $\zeta(i)$ (by definition at an endpoint of $\zeta(i) \cap \partial(\overline{R})$) and, in addition, the other endpoint of $\zeta(i) \cap \partial(R)$ is not in Q. Even if we cannot determine it from our observations alone, our original definition of the ordering (depending on P) is still valid; because the new disk has its center on the neighbor-feasible arc of C^*_i, it must be a neighbor of C^*_i. Furthermore, since it is the other (further clockwise around the boundary of $\overline{R}$) endpoint of $\zeta(i) \cap \partial(R)$, the remaining set of points at which C^*_i can be incident to P, which is a subset of $\zeta(i) \cap \partial(R)$, is counterclockwise from all points of the new disk (around the boundary of $\overline{R}$). Hence, the new disk cannot be between C^*_{i-1}, C^*_i and can be inserted between C^*_i, C^*_{i+1}.

The remainder of the updates involve updating V_c and E_c, and in turn updating Q to not include confirmed vertices or edges; as any vertex or line is automatically confirmed when three p-disks are tangent to it, and thus these checks remain in constant time. Updating the relevant stored information is constant for each element of R, Q, Q^*, V_c, E_c and S_{disks} we update, and for each set only a bounded number of elements (the neighbors of the probed point) are updated, so the total updating time has complexity $O(1)$ per probe.

5.2.3 Example

These concepts are shown in Figure 5.3. The vertices of the polygon are labeled in clockwise order, with v_1 being the acute angle vertex; the edges are labeled in clockwise order as specified previously. The main features are represented as follows: cp-disks are shown by blue circles, and p-disks which have been condensed are shown by yellow circles; the infeasible region R is green;

The panels show: *(a)* P and D before any probes; *(b)* the initial probe x_0 on the boundary of D and x_1 is one of the intersection points of the disk resulting from probing x_0 and $\partial(D)$; *(c)* illustration of all but one of the probes chosen during Phase 1 of the algorithm; *(d)* after 7 probes the edge e_1 of P is *confirmed* by the algorithm, and the disks incident to it are condensed; *(e)* in Phase 2, case (c) of the algorithm occurs, resulting in a probe at x_7. This confirms v_1, and therefore we condense the disks which are incident to it (centered at x_2 and x_7); *(f)* after 14 probes, all the edges and vertices are confirmed, and so P has been determined.

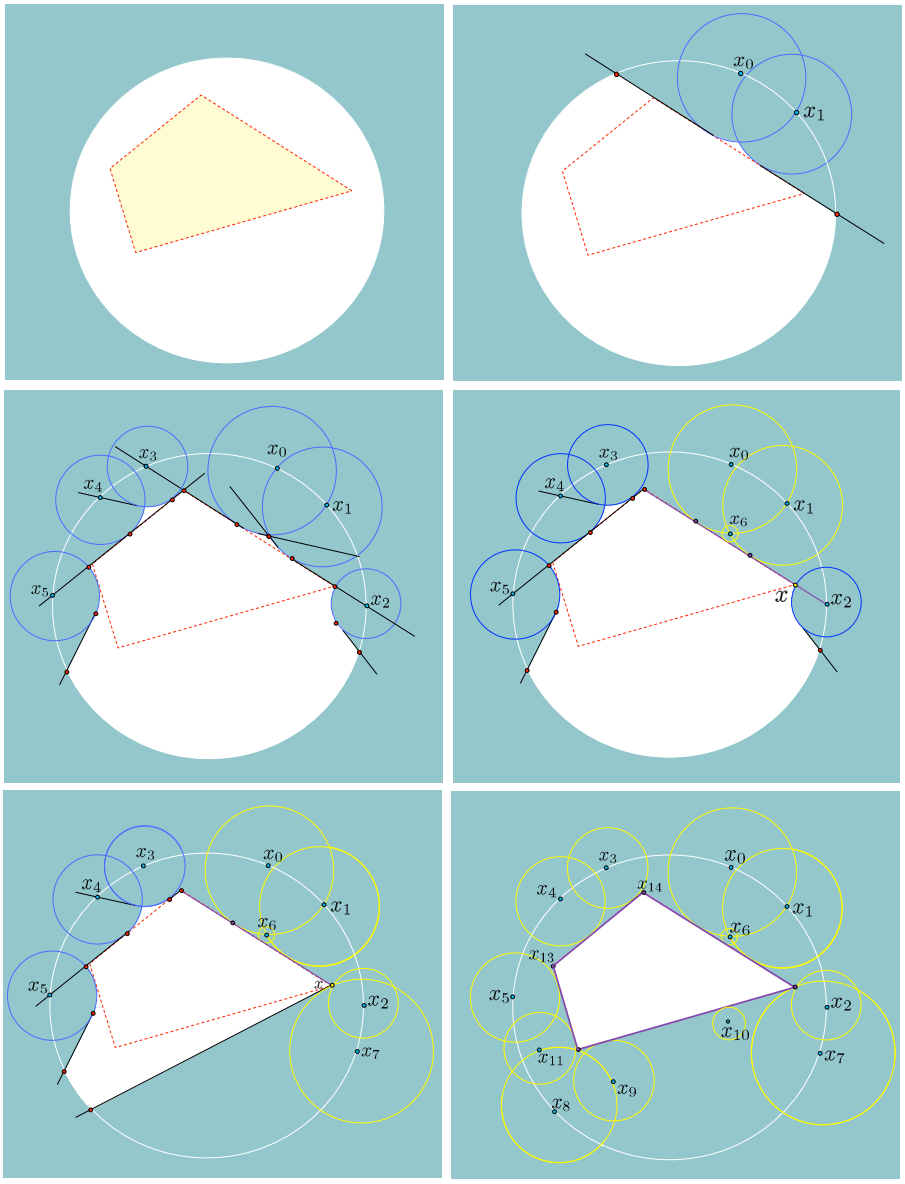

Figure 5.3: An illustration of Algorithm 3 probing to determine the shape and location of a polygon with $n = 4$ vertices and $k = 1$ acute angle. This instance is solved with 14 ($\leq 3.5n + k + 2$) probes.

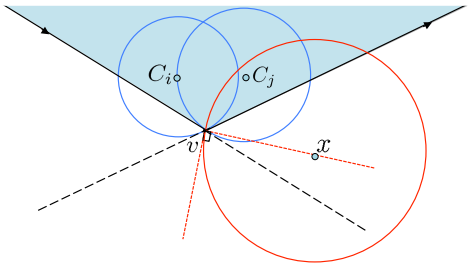

Figure 5.4: If v is a confirmed vertex and the disk resulting from $x \in R$ is tangent to v, then $\angle_P(v)$ is acute while $\angle_R(v)$ is not acute.

5.3 Complexity bounds on the algorithm

We first establish the following notation. Let v be a vertex of P; we then write $\angle_P(v)$ to refer to the angle of P at v. If v is confirmed, we note that this means the algorithm would have condensed the disks incident to v, so that $\overline{R}$ would have an angle at v; we write $\angle_R(v)$ to refer to this angle.

Note that $\angle_P(v)$ is always contained in $\angle_R(v)$ and that $\angle_R(v)$ never increases as the algorithm goes on.

5.3.1 Preliminary lemmas

We present four preliminary geometric results, which we will be using to show our main theorems.

Lemma 5.3.1. *Assume that v is the intersection point on $\partial(R)$ of the boundaries of two p-disks C_i and C_j, neither of which contains the other. If we probe from $x \in \overline{R}$ such that $x \neq v$, the resulting p-disk C cannot pass through v unless $\angle_P(v)$ is acute. If $\angle_P(v)$ is acute and C passes through v, then $\angle_R(v)$ becomes acute.*

Lemma 5.3.2. *Let v be a confirmed vertex of P, and let $x \in \overline{R}$ be the next probed point which produces a disk C Note that v is already confirmed, so $x \neq v$ since we don't probe confirmed vertices. Then C can be incident to v only if $\angle_P(v)$ is acute, $\angle_R(v)$ is not acute; furthermore, afterwards, $\angle_R(v)$ will be acute (so no new p-disk can be incident to v).*

Corollary 5.3.3. *Let v be a vertex of P such that when v is confirmed, it is* not *by being probed directly. Then,*

- *if $\angle_P(v)$ is not acute, when the algorithm finishes the number of p-disks incident to it is at most 2*
- *if $\angle_P(v)$ is acute, when the algorithm finishes the number of p-disks incident to it as at most 3*

This corollary follows directly from the preceding lemma.

Lemma 5.3.4. *Let e be a confirmed edge and v be one of its endpoints. Let $x \in \overline{R}$ such that x doesn't lie on the line extending e. If we probe from x, the resulting disk cannot be incident to v unless v is an acute angle vertex of P.*

We also note that as long as we only probe from points in $\partial(R)$ which are not confirmed vertices or in the interior of any line segment on $\partial(R)$ contained by a confirmed line, we will never create a p-disk which will be incident to the interior of any previously confirmed edge.

5.3.2 Undesirable confirmations

The bounds derived in the previous section are only violated (by 1) if v is confirmed while incident to three p-disks, one of which is the zero-disk centered at v itself (this applies regardless of whether $\angle_P(v)$ is acute). However, we note that if one of the two non-zero p-disks is also tangent to one of the edges of P adjacent to v, we may associate it with that edge instead (so that the bound is not considered violated), and hence need only worry about the possibility that neither of the non-zero p-disks are tangent to an adjacent edge. We call such cases *undesirable confirmations*, as each occurrence increases our bound on the number of probes needed to confirm P.

Lemma 5.3.5. *Let C be a disk centered at x and v be a point. Let l_x be the ray with source at v and passing through x. Let y be the first intersection point of L and $\partial(C)$. In addition, let l be the ray with source at v and tangent to C lying to the left of l_x, and let z be the point where l is tangent to C. Finally, let x' be any point on the arc of $\partial(C)$, and C' be a disk centered at x'. Then the ray l' with source at v and tangent to C' (such that C' is to the right of l', as in Figure 5.5) is tangent to C' outside C.*

Proof. Let u_x and u' be the intersection points of $L(C, x')$ with l_x and l', respectively. If we consider the triangle with vertices at v, u_x, u', it is clear that the inner angle of $\angle u$ is obtuse, and hence the inner angle of $\angle u'$ is acute. Therefore, the perpendicular line from x' to l' intersects l' outside of C, i.e., the tangent point of l' to C' lies outside of C. □

Lemma 5.3.6. *In Phase 2 of Algorithm 3, if at any step our probe was from case (b) (where the ray l intersects some other straight-line piece L of the boundary of R), where the angle between L and l is is not acute, and the probe does not confirm a vertex, our next probe will also be of case (b), and also with a non-acute angle.*

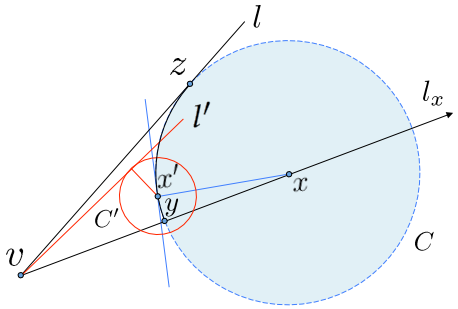

Figure 5.5: The ray l' with source at v and tangent to C' cannot be tangent to C' inside of C, where C' is a disk centered at an arbitrary point lying on $\partial(C)$ and between y and z.

Proof. Case (b) of Phase 2 of the algorithm concerns the case where the ray l on the current 'last' edge e_t intersects some (confirmed or unconfirmed) line L on the boundary of R. We let C_i^* be the first cp-disks, in the clockwise order, on L; then C_i^* has L on the boundary of its neighbor-shadow set. We then note that C_{i-1}^* is a condensed disks which is by definition the last point, in the clockwise order, which we know exists on the edge e_t. Let x be, as in the algorithm, the intersection point of L and l which we are probing, and let C be the disk which is produced from probing x.

We now consider the line $L_{i-1,i}$; suppose that C is *not* interior-disjoint with $L_{i-1,i}$. In that case, we have a contradiction as no convex P can be incident to C_{i-1}^* and C_i^* without passing through the interior of C. Therefore, C must be interior-disjoint with $L_{i-1,i}$ and can be inserted into the ordered list S_{disks} between C_{i-1}^* and C_i^*; but then this means that the ns-set of C has a ray of its boundary which intersects l before C does (since the angle between l and L is not acute, the ray on the boundary of the ns-set intersects l rather than simply lying entirely on one side of l). Thus, unless our algorithm passes to case (a) or updates t (which both require confirming a new vertex), the next probe will be of case (b) (and also with a non-acute angle, since the angle in this case is strictly increasing). □

We note that an easy corollary of this lemma is that if case (b) occurs at a non-acute angle, it will continue until a new vertex is confirmed; since undesirable confirmations by definition cannot happen in case (b), the next confirmed vertex cannot be undesirable. We may now use this to present a bound on the number of undesirable confirmations.

Lemma 5.3.7. *At most one undesirable confirmation occurs during Phase 1.*

Proof. First, since we only probe from Q during Phase 1, any probe which returns a zero-disk must confirm a vertex, as no point in Q can correspond to the interior of an edge of P. In this case, by definition, an undesirable confirmation occurs if and only if we probe from a point x which is in the intersection of two p-disks and receive $f_P(x) = 0$; therefore, by definition, if we probe from Q^* we will not get an undesirable confirmation.

Suppose that we have one undesirable confirmation so far, for vertex v of P, and suppose we are still in Phase 1; therefore, E_c is empty. In particular, this means that segments of the lines produced by the condense operation on disks incident to v are on the boundary of R. Given one of these segments, we see that one endpoint will be v and the other endpoint cannot be a confirmed vertex (since otherwise we could confirm the line and go to Phase 2); furthermore, the other endpoint cannot be incident to two p-disks either since we would still be able to confirm the line in question. Therefore, the other endpoint will be in Q^*, so Q^* will not be empty. Therefore, once one undesirable confirmation occurs in Phase 1, there cannot be another until Phase 2 begins. □

Lemma 5.3.8. *Let m be the number of undesirable confirmations which occur over the course of the algorithm. Then $m \leq n/2 + 1$.*

Proof. We first wish to show that in Phase 2 only every other confirmed vertex can be an undesirable case. More specifically, we assume v is confirmed by an undesirable confirmation, and show that the next vertex to be confirmed cannot be an undesirable confirmation. We let l and e_t be defined as in the algorithm. We also assume to the contrary that the next vertex confirmation after v is undesirable.

Since v is an undesirable confirmation, it must have been confirmed in case (c), where the endpoint of l on the boundary of R falls on the intersection of the boundaries of two cp-disks. When v is confirmed, these two cp-disks are condensed, and the next probe must be of case (b), at the intersection between l and one of the lines L produced through v by the condense operation. We let this intersection be x_i and the resulting cp-disk be C_i^* (Figure 5.6).

If any probe afterwards (before the confirmation of the next vertex, which we have assumed to be undesirable) falls on l, it cannot be of case (c) and cannot be the assumed undesirable next confirmation; therefore, it was generated by case (b), as the intersection of l and a line on $\partial(R)$ tangent to C_i^*. However, since the center of C_i^* is on l, the angle at this intersection cannot be acute (Figure 5.6(a)), and therefore by Lemma 5.3.6, the next confirmed vertex cannot be an undesirable case, thus producing a contradiction.

Therefore, we may assume that x_i is the last probe to fall on l before the next vertex is confirmed; thus, every probe between x_i and the next confir-

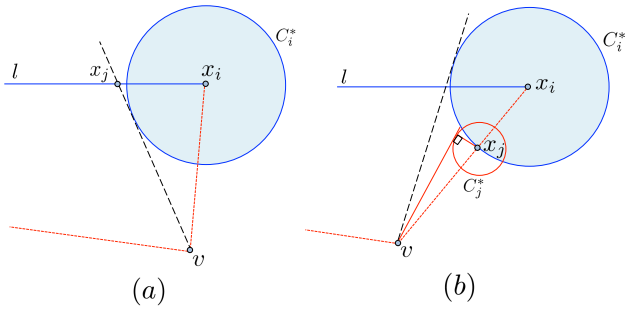

Figure 5.6: v is the vertex identified by an undesirable confirmation and C_i^* is the p-disk generated by the subsequent probe. (a) Any probe which falls on l before the next undesirable confirmation of a vertex must be the intersection of l and a line on $\partial(R)$ where the angle at the intersection is not acute, and must be the result of case (b) of Algorithm 3. (b) Any probe after x_i which is the last probe lying on l falls on the intersection of ζ_i and a tangent line.

mation is of case (c) (since an occurrence of case (a) would require the next confirmation to happen first, and case (b) by definition always falls on l). It is clear by a simple inductive argument that every probe between x_i and the new undesirably-confirmed vertex falls on the boundary of C_i^*: in case (c), the next probed point will be on the other endpoint of the nf-arc $\zeta(i)$. If the resulting p-disk intersects l outside of C_i^*, then we have a contradiction as C_i^* can immediately be shown to be disjoint with P, which by definition is not possible, so C_i^* must remain the cp-disk l intersects; but then, since we have assumed that every probe until the next vertex confirmation is of case (c), it must remain on the boundary of C_i^* until the next vertex is confirmed.

We now note that each new probe which does not confirm a new vertex decreases $\zeta(i)$. We note as well that because x_i is located where one of the lines associated with v (when it is confirmed) intersects l, $\zeta(i)$ by definition is initially the arc between l and the line joining v and x_i (Figure 5.6(b)); we let this arc of C_i^* be called ζ^*.

We now consider the probe immediately preceding the probe which results in the undesirable confirmation. By the above, this must occur on ζ^*; but then by Lemma 5.3.5, after this probe neither of $\zeta(i)$'s endpoints are intersections of the boundaries of two cp-disks. But this immediately implies that the next probe, which is at an endpoint of $\zeta(i)$, cannot be an undesirable confirmation. Hence, since none of these probes can be an undesirable confirmation, the next vertex confirmation will not be undesirable, contradicting our assumption that it is.

Thus, we have proved that in Phase 2, whenever an undesirable confirmation occurs, the next vertex to be confirmed cannot be an undesirable confirmation. Since we have already proved that in Phase 1 there can be at most one undesirable confirmation, there are in total at most $n/2 + 1$. □

5.3.3 Bounding the number of probes used

We now wish to find an upper bound for the number of probes used by Algorithm 3; this is achieved by analyzing the number of p-disks that can be incident to any edge or vertex of P when it is confirmed. We now assume that no undesirable confirmation occurs; we will then note that since each undesirable confirmation adds at most one probe, and by Lemma 5.3.8 there are at most $n/2 + 1$, we can add this to the bound we derived to obtain the true bound.

At any given step in the algorithm, let $\phi(e)$ and $\phi(v)$ denote the number of p-disks incident to unconfirmed edge e and unconfirmed vertex v respectively; and let $\omega(e)$ and $\omega(v)$ denote the number of p-disks which are incident to confirmed edge e and confirmed vertex v, respectively.

We first consider the number of p-disks any object can have adjacent to it at the moment it is first confirmed; by convention, if a p-disk is incident to both some confirmed vertex and some confirmed line, we associate it with the vertex only. We perform this analysis on the two basic phases of the algorithm.

For Phase 1 (i.e., confirming the first edge), there are two possible cases for the number of probes which will suffice to confirm the first edge e_1 with clockwise endpoint v_1:

- If $\phi(v_1) \leq 1$ three disks are sufficient to confirm e_1.
- If v_1 is confirmed or $\phi(v_1) = 2$, then two disks are sufficient to confirm e_1.

We will conduct the same analysis for Phase 2 by computing the possible values of $\omega(v_{i-1})$ and $\omega(e_i)$ when they are first confirmed (which depends on whether v_{i-1} is acute or not) for $1 < i \leq n$. We note that no vertex can be confirmed on $\partial(D)$ because $P \in \text{Int}(D)$.

Case 1: v_{i-1} is not confirmed and $\phi(v_i) \leq 1$. Since v_{i-1} is not confirmed but e_{i-1} is confirmed, $\phi(v_{i-1}) \leq 1$. We consider the two possible sub-cases: either v_{i-1} is not an acute angle vertex of P, or it is.

- Suppose v_{i-1} is not an acute angle vertex. It could either have been confirmed by case (b) or case (c) from Phase 2 of the algorithm.

– Suppose it was confirmed by case (b); let x be the point probed. For case (b) of the algorithm to confirm a vertex, the result of the probe must be 0 (i.e., $f_P(x) = 0$), and this new zero-disk is the only disk incident to v_{i-1}; thus $\omega(v_{i-1}) = 1$. In this case, x (which is actually v_{i-1}) cannot lie on the boundary of D (as in this case $x \in P \subset \text{Int}(D)$), so x is on a segment of an (confirmed or unconfirmed) line L on $\partial(R)$; this line will then be confirmed as e_i with $\omega(e_i) = 2$.

– Suppose it was confirmed by case (c). By Lemma 5.3.4, the new p-disk cannot pass through v_{i-1}, so $\omega(v_{i-1}) = 1$. We observe that to confirm v_{i-1}, the new p-disk must reduce the feasible arc of the previous p-disk containing v_{i-1} to a single point; to do this, it must confirm e_i. Hence, since $\omega(v_{i-1}) = 1$ and $\phi(v_i) \leq 1$, we get $\omega(e_i) = 2$.

Therefore, in all cases, $\omega(v_{i-1}) = 1$ and $\omega(e_i) = 2$.

- If v_{i-1} is an acute angle vertex. This is similar to the above case, except that as Lemma 5.3.4 doesn't hold for acute angles, we include the possibility that in case (c) the resulting p-disk will pass through v_{i-1}. If so, v_{i-1} is confirmed, and the next iteration of the algorithm will be case (a). As $\phi(v_i) \leq 1$, $\omega(e_i) = 2$, and when v_{i-1} is confirmed in the next iteration $\omega(v_{i-1}) \leq 2$.

Case 2: v_{i-1} is not confirmed and either $\phi(v_i) = 2$ or v_i is confirmed. This case is similar to case 1, except that because $\phi(v_i) = 2$ (or v_i is confirmed), e_i is incident to at most one disk, and v_{i-1} will be confirmed immediately after e_i is confirmed. So, $\omega(e_i) = 1$ and $\omega(v_{i-1}) \leq 2$, if v_{i-1} is acute and $\omega(v_{i-1}) = 1$ if it is not.

Case 3: v_{i-1} is confirmed and $0 \leq \phi(v_i) \leq 1$. We consider the two possible sub-cases: either v_{i-1} is not an acute angle vertex of P, or it is.

- v_{i-1} is not an acute angle vertex. Since v_{i-1} is confirmed before e_i and v_{i-1} is not an acute angle, by Lemma 5.3.4, $\omega(v_{i-1}) = 2$, and case (a) will immediately follow in the algorithm. The next edge e_i will be confirmed by two incident disks since $\phi(v_i) \leq 1$, so $\omega(e_i) = 2$.

- v_{i-1} is an acute angle vertex. According to Lemma 5.3.4, it is possible that v_{i-1} has been confirmed with three disks as v_{i-1} is an acute angle. Therefore, $\omega(v_{i-1}) \leq 3$. As in the previous case, $\omega(e_i) = 2$.

Case 4: v_{i-1} is confirmed and either $\phi(v_i) = 2$ or v_i is confirmed. We again consider the same two possible sub-cases as in the above cases.

- v_{i-1} is not an acute angle vertex. As in case 3, $\omega(v_{i-1}) = 2$, but the next edge will be confirmed with one incident disks since v_i is incident to more than one disk (or already confirmed), so $\omega(e_i) = 1$

- v_{i-1} is an acute angle vertex. As in case 3, $\omega(v_{i-1}) \leq 3$, and $\omega(e_i) = 1$ since v_i is incident to multiple disks (or already confirmed).

Finally, it is clear that v_n will be confirmed with one disk. Table 5.1 summarizes the result for the above four cases.

Theorem 5.3.9. *Our algorithm uses at most* $3n + m + k + 1 \leq 3.5n + k + 2$ *probes to find* P*, where* $k \leq 3$ *is the number of acute angles of* P*; each probe is computed in* $O(1)$ *time, thus leading to an overall time complexity of* $O(n)$*.*

Table 5.1: the number of incident p-disks to v_{i-1}, e_i for $(1 < i \leq n)$

Case	v_{i-1}	v_i	v_{i-1}: Not acute $\omega(v_{i-1}), \omega(e_i)$	v_{i-1}: acute $\omega(v_{i-1}), \omega(e_i)$
1	NC	NC, $\phi(v_i) \leq 1$	$1, 2$	$\leq 2, 2$
2	NC	C or $\phi(v_i) = 2$	$1, 1$	$\leq 2, 1$
3	C	NC, $\phi(v_i) \leq 1$	$2, 2$	$\leq 3, 2$
4	C	C or $\phi(v_i) = 2$	$2, 1$	$\leq 3, 1$

Proof. We note that no p-disk generated at any point by the algorithm can be incident to a previously-confirmed edge or to a previously-confirmed non-acute angle vertex once both edges adjacent to it have been confirmed. Note also that since the algorithm never probes from the interior of $\overline{R}$, the algorithm never uses a probe which returns -1. Therefore, the number of probes needed is equal to the sum of the number of p-disks incident to each edge and vertex of P when they are confirmed, with the possible additional k for the acute angles already taken care of by assuming the worst case at time of confirmation. Let n_j be the number of times case j occurs, and k_j be the number of times case j occurs with an acute vertex; then $\sum_{j=1}^{4} n_j = n - 1$ and $\sum_{j=1}^{4} k_j \leq k$ since the cases begin once e_1 is confirmed.

We now consider the number of p-disks incident to each edge and vertex of P when they are confirmed, assuming no undesirable confirmations:

- e_1 is incident to at most 3 p-disks when it is confirmed

- For $j = 1, 4$, by Table 5.1 we note that $\omega(v_{i-1}) + \omega(e_i) \leq 4$ if v_{i-1} is acute, and $\omega(v_{i-1}) + \omega(e_i) = 3$; hence at most $3n_j + k_j$ probes were used.
- For $j = 2$, by Table 5.1 we note that $\omega(v_{i-1}) + \omega(e_i) \leq 3$ if v_{i-1} is acute, and $\omega(v_{i-1}) + \omega(e_i) = 2$; hence at most $2n_2 + k_2$ probes were used
- For $j = 3$, by Table 5.1 we note that $\omega(v_{i-1}) + \omega(e_i) \leq 5$ if v_{i-1} is acute, and $\omega(v_{i-1}) + \omega(e_i) = 4$; hence at most $4n_3 + k_3$ probes were used

Consider what happens in case 3 (with vertex v_{i-1} and edge e_i); it occurs when v_{i-1} is incident to two disks (or is confirmed) before e_i is confirmed. If $i = 2$, then e_1 must have been adjacent to 2 p-disks. If $i > 2$, then case 3 was preceded by either case 2 or case 4; if it was case 4, then since v_{i-1} was already confirmed, e_{i-1} must have been confirmed with one fewer p-disk than our above bounds.

Thus, every instance of case 3 (which requires one more probe per vertex-edge pair than cases 1 or 4), there is a corresponding instance either of case 2 (which requires one fewer probe per vertex-edge pair than cases 1 or 4) or of case 4 (or the base case) in which at least one fewer probe was used than the bound above. So, since case 3 is the only case in which more probes are required than cases 1 and 4, and since we showed that every instance of case 3 is 'offset', we can bound the total number of probes needed by the number needed if only cases 1 and 4 occurred.

Thus, the pairs $(v_1, e_2), ..., (v_{n-1}, e_n)$ plus e_1 require at most $3n + k$ probes to confirm; the final vertex v_n requires one more, giving an upper bound of $3n + k + 1$ probes with the assumption that no undesirable confirmations occurred. Each undesirable case increases the upper bound by at most 1, and the number of such cases (by Lemma 5.3.8) is $m \leq n/2 + 1$. Hence, we compute our true upper bound as $3n + m + k + 1 \leq 3.5n + k + 2$ probes. Finally, we note that in Section III(B) we showed that each probe requires $O(1)$ computations, and therefore the total time required by the algorithm is $O(n)$. □

5.4 Problem 2: Identifying a convex polygon from a known set

We now consider the problem of identifying a convex polygon P with n vertices from a known finite set. Let Γ be a (known) finite set of convex polygons and D be a (known) disk. We are now asked to identify P using as few probes as possible, knowing that $P \subset \text{Int}(D)$ and that $P \in \Gamma$ (where a polygon is considered a member of Γ if it can be rotated and translated to match an element of Γ). Let $m = |\Gamma|$ and n' be the maximal number of vertices on any

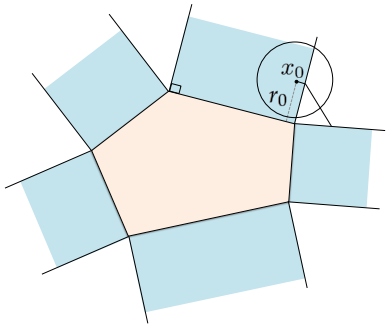

Figure 5.7: The partitioning of the exterior of a polygon. Any probes in the same part are incident to a common vertex or edge.

polygon in Γ. We show that $2n + 2$ probes are sufficient to find P with a time-complexity of $O(1)$ per probe once a $O(n'm)$ pre-processing phase has been completed; this allows the algorithm to find efficient use if many polygons P must be identified from the same Γ in addition to reducing the number of probes.

Remark: Since our algorithm will proceed by confirming the edges of P in counterclockwise order, P is *not* considered to be in Γ if a reflection is required to produce a match to an element. However, if we wish to include reflections, we can run our algorithm on an augmented set consisting of Γ plus the reflections of all elements of Γ; this does not increase the number of probes needed to find P by the result above, but it doubles the computation time needed for the algorithm.

Let e_{min} and e_{max} be the minimum and maximum, respectively, over lengths of all edges in Γ, and ψ_{max} be the maximum over all angles, of the polygons in Γ.

Lemma 5.4.1. *Let $x_0 \notin D$ be a point and C_0 be the p-disk (of radius $r_0 > 0$) generated by probing x_0. Let x_1 be another point such that $\mathrm{dist}(x_1, x_0) < d$ where $d = \min\{e_{min}, r_0 \sin(\frac{\pi - \psi_{max}}{2})\}$. Let C_1 be the p-disk generated by probing x_1. Then:*

- *If C_0 is incident to an edge, C_1 will be incident to the same edge or one of its endpoints.*
- *If C_0 is incident to a vertex, C_1 will be incident to the same vertex or one of its adjacent edges.*

Proof. Consider all the perpendicular half-lines of the edges of P rooted at its vertices and contained in the exterior of P. These lines partition the exterior

of P to $2n$ regions. Note that if we probe from any point inside one of these regions, the resulting p-disk hits the edge or vertex bordering the region (Figure 5.7). After probing from x_0, if we choose x_1 inside the disk centered at x_0 with radius d, it is easy to see that x_1 lies in the same region that x_0 lies in, or in an adjacent region. □

Lemma 5.4.2. *Let C_0 and C_1 be two intersecting disks centered at x_0 and x_1, respectively. Let $p_{0,1}$ be an intersection point of their boundaries . If there are two intersection points, let $p_{0,1}$ be the one closer to $L_{0,1}$. For $i = 0, 1$, let p_i be the point where $L_{0,1}$ is tangent to C_i. Then,* $\text{dist}(p_0, p_{0,1}) < \text{dist}(x_0, x_1)$.

Proof. Let $\angle p_0 x_0 p_{0,1} = \alpha$ and let $\angle p_1 x_1 p_{0,1} = \beta$ where p_1 is the tangent point of $L_{0,1}$ on C_1^*. It is clear that $\alpha + \beta \leq \pi$. (Note that $\alpha + \beta = \pi$ when there is only one intersection point.) $\angle p_0 p_{0,1} p_1 = \pi - (\alpha + \beta)/2 \geq \pi/2$. Thus, the triangle $p_{0,1} p_0 p_1$ is obtuse and $\text{dist}(p_0, p_{0,1}) < \text{dist}(p_0, p_1)$. Since $\text{dist}(p_0, p_1) \leq \text{dist}(x_0, x_1)$ (since p_0, p_1 are projections of x_0, x_1 on $L_{0,1}$), $\text{dist}(p_0, p_{0,1}) < \text{dist}(x_0, x_1)$. □

We now present an algorithm as a series of lemmas. Our algorithm maintains a list of vertices and edges confirmed, as well as a point on e_1 (after it is confirmed). At each step (after initialization, which confirms the first edge), we then take the last vertex or edge confirmed (in clockwise order) and show that it is possible to determine the next edge or vertex with a single probe. This allows us to uniquely determine P using $2n + 2$ probes (one probe per vertex or edge, plus initialization cost of 2 probes). Lemma 5.4.3 shows that either the first edge can be confirmed with 3 probes, or the first edge and first vertex can be confirmed with 4 probes; Lemma 5.4.4 shows that afterwards, the next edge or vertex can be determined using only one probe. Theorem 5.4.5 then puts them together to obtain our upper bound on the number of probes needed.

Lemma 5.4.3. *There is an approach that can find either the first edge of P with three disks or find the first edge and one of its endpoints using 4 probes.*

Proof. For any point x_i, let C_i be the p-disk (with radius r_i) generated by probing it. First we probe from a point $x_0 \in \partial(D)$. Choose $x_1 \in \partial(D)$ such that $\text{dist}(x_1, x_0) < d$ where $d = \min\{e_{min}, r_0 \sin(\frac{\pi - \psi_{max}}{2})\}$. By Lemma 5.4.1, if we probe from x_1, C_1 and C_0 are incident with the same edge, the same vertex or an edge and its endpoint. Since $d \leq r_0$, the resulting disks are not disjoint. Let $p_{0,1}$ be the intersection of ζ_0 and ζ_1. Without loss of generality, assume that $r_0 \leq r_1$. We note that it is not possible for C_1 to contain C_0 completely (if $C_0 \subset C_1$, then x_0, x_1 and $p_{0,1}$ become collinear while $p_{0,1}$ has to be a point of P; however, since x_0 and x_1 lie on $\partial(D)$, $p_{0,1}$ lies outside of D which is not possible and thus produces a contradiction). Thus, we can choose x_2 to be

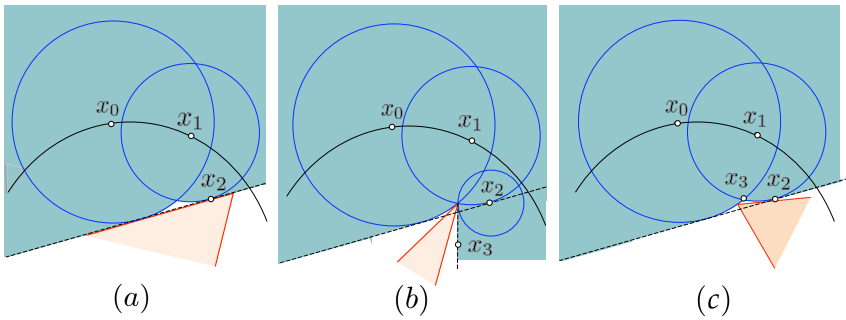

Figure 5.8: The third probe from x_2 results in the disk C_2. There are three possible cases: (a) If C_2 is a zero disk, then three probes are sufficient to confirm the first edge. (b) If $\partial(C_2)$ intersects the intersection of $\partial(C_0), \partial(C_1)$, the intersection point is confirmed as a vertex and probing from x_3 will be tangent to the adjacent edge. (c) Otherwise, probing from x_3 (the intersection point of $\partial(C_0), \partial(C_1)$) will confirm an edge and one of its endpoints.

the intersection of $L_{0,1}$ and C_1. Note that the intersection of $\partial(C_1)$ and $\partial(C_2)$ has to lie on ζ_1.

- If $r_2 = 0$, then $L_{0,1}$ is confirmed as an edge (Figure 5.8(a)).
- If $r_2 = \text{dist}(x_2, p_{0,1})$ then $p_{0,1}$ is confirmed as a vertex. Call this confirmed vertex v. Choose $x_3 \in \partial(R)$ on $L_2(v)$ such that $\text{dist}(x_3, v) < e_{min}$. Note that it is not possible for C_3 to intersect v. The tangent line of C_3 from v is confirmed as the first edge (Figure 5.8(b)).
- If $0 < r_2 < \text{dist}(x_2, p_{0,1})$, then we choose x_3 on $p_{0,1}$. Note that according to Lemma 5.4.2, $\text{dist}(x_2, x_3) < d$ (Figure 5.8(c)).
  - If $r_3 = 0$, $p_{0,1}$ is confirmed as a vertex. By Lemma 5.4.1, the tangent line of C_2 from v is confirmed as its incident edge.
  - If $r_3 > 0$ then C_3 must be tangent to $L_{1,2}$. Then, $L_{1,2}$ is confirmed as a line and its intersection with C_0 is confirmed as its endpoint.

□

Lemma 5.4.4. *Let $e_1, ..., e_n$ be the edges of P in counterclockwise order. Suppose we have a line passing through e_1 and a point p lying on the interior of e_1. Let v_n and v_1 be the unknown endpoints of e_1. We can find v_1 using a single probe.*

Proof. Without loss of generality assume that the given line is horizontal and v_n is the left endpoint of e_1. First, suppose that v_n is given. Based on the edge

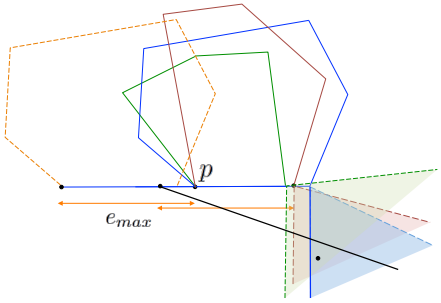

Figure 5.9: The intersection of the corresponding regions to all candidate vertices.

lengths in Γ, there are a finite number of candidate vertices for v_1. Each of the candidate vertices is associated with a region (determined by the polygon which the candidate is based on) from which a p-disk will pass through it. If we probe from a point which in the intersection of all these regions, the disk will hit v_1. This intersection is not empty and is formed by the vertical line which passes through the rightmost candidate vertex and the perpendicular line to its next incident edge which has the lowest intersection point with the rightmost vertical line (Figure 5.9).

Consider the case in which v_n is not given; we bound its location on the given line. Note that $|e_1| \leq e_{max}$, so v_n lies between p and p' where $p' \in D$ is the leftmost point such that $\text{dist}(p, p') \leq e_{max}$. Again, we aim to find a region which is the intersection of all regions for the candidate vertices when v_1 can lie between p and p'. This region is encompassed between two lines. One is the vertical line that passes through the rightmost candidate vertex, assuming that v_1 lies on p. The other is the line perpendicular to the next incident edge of e_1 which has the lowest intersection point with this vertical line assuming that v_n lies on p' (Figure 5.9). □

Theorem 5.4.5. *$2n + 2$ probes are sufficient to determine a convex polygon P from a set of models Γ, with $O(n'm)$ time complexity for preprocessing and $O(1)$ for each probe.*

Proof. By Lemma 5.4.3, to determine the first edge and a point on it, either 3 probes are sufficient or 4 probes are used and one of its endpoints is confirmed as well. By Lemma 5.4.4, the endpoint of the edge can be determined by a single probe. To find the next edge incident to the endpoint, a single probe can be used from a point which lies on the extension of the current

edge from its determined endpoint with the distance less than e_{min}. The p-disk generated by this probe will be guaranteed to hit the next edge. Using this strategy, we need $3 + n + (n - 1)$ probes to identify P.

In order to do this, we must first compute and the e_{max}, e_{min} and ψ_{max}, as well as the region, which requires $O(n'm)$ time. Finding the shape of the region to probe from in the proof of Lemma 5.4.4 (relative to an arbitrary line) requires computing the lowest intersection point of $O(n'm)$ lines with a vertical line; once this is done, finding our next probe point at each step can be achieved in constant time. The incident edge of a vertex can be determined in $O(1)$. Thus, the time complexity of the algorithm is $O(n'm)$ for the preprocessing of Γ, $O(1)$ for each probe and the complexity of a query is $O(n)$. □

5.5 Conclusion

In this work, we addressed geometric probing with a type of proximity probe and presented algorithms to solve two associated variations of the related problems. Algorithm 3 determines the shape an unknown convex polygon P (with n vertices, $k \leq 3$ of which are acute angle vertices) requiring at most $3.5n + k + 2$ probes, with the position of each probe requiring $O(1)$ time to compute. It can be seen that $O(n)$ probes with $O(1)$ computations per probe is optimal, and thus any improvements to our algorithm must focus purely on reducing the constants and overhead costs.

We addressed the problem introduced in [120] to identify P from among a finite set Γ, and solved it using at most $2n + 2$ probes and constant computations per probe once an $O(n'm)$ preprocessing routine is completed, where m is the size of Γ and n' is the maximal size of the polygons in Γ. Thus, while Algorithm 3 focused on reducing both the number of probes and the amount of computation needed per probe, the algorithm uses the extra information provided by Γ primarily to optimize the number of probes. However, if the problem is repeated for many P and one Γ, the algorithm benefits from only needing to preprocess Γ once.

There are many intriguing directions for future foundational work expanding on these results. For example, although our algorithms have optimal order of growth and a reasonable coefficient for the number of probes, it would be interesting to determine the true minimum number of probes needed to determine that an n-sided convex polygon P. Another very important question, particularly for applications, is how to model and develop algorithms for the case where probe measurements contain errors or noise; instead of precisely identifying the unknown polygon, the goal here would be to give a sufficiently accurate approximation of it. Finally, it is natural to

ask whether our methods could be extended to higher dimensions or for non-convex polygons (as Boissonnat and Yvinec's work did for finger probes [33]).

Part IV

Closing remarks

Chapter 6

Conclusion

"We can only see a short distance ahead, but we can see plenty there that needs to be done."

- Alan Turing

THIS final chapter summarizes the main results of the thesis and describes possible ways that our results can be extended or improved in future work. In this thesis, we have seen how approaches from computational geometry can be applied to a variety of research problems in the field of robotics and automation. We have focused on two topics in this field: orienting and probing. The first problem investigates how to orient industrial parts of the same geometry into a unique final orientation when the initial orientation is not known. The probing problem tries to measure the geometric shape and orientation of an unknown object.

6.1 Orienting parts

In Part II of this thesis, we have discussed the problem of orienting (or feeding) parts. From a practical point of view, orienting parts is a crucial task for the throughput of an assembly line. In this work, we have focused on three questions related to part orienting which are described in this section.

Existing theoretical research on the problem of part orienting from a computational geometric point of view, considers abstract and restricted instances of the orienting problem such as planar, convex, polygonal and perfect parts

without any shape variation. The work does not also consider the dynamics of friction and collisions. Although there has been a great deal of research into these problems, the various complex physical aspects which arise in practice mean that there remain many challenges and open questions. A major direction of our work and of future work on the topic is the use of more realistic assumptions, so that the resulting algorithms and designs are more reliable in practice.

One of the assumptions in many of the proposed designs for part orienting is that the parts are considered to be perfectly identical. However, while convenient, this is not realistic; manufacturing and assembly processes produce parts that vary in size and form which result in slight imperfections in their geometric shape. The few papers that assume imperfect parts generally assume a very restrictive model for shape variations. In these models, parts are often considered as a convex polygon whose vertices are known to be within given circles or rectangles. In this thesis, we have considered a more general model for shape variations that allows variation along entire boundary of an object, both in two and three dimensional space. Based on this model of shape variation, we have studied the problem of orienting planar parts. Our proposed model considers a family of parts defined by an outer and an inner envelope whose set difference contains the boundary of the nominal part. These envelopes are assumed to be closed semi-algebraic sets with a total of n boundary features. This model for shape variation can be studied for different approaches of part feeding such as fence/trap design as well as other manipulation tasks. Another intriguing direction for future work is to consider this model of shape variation not only for the part but also for the actuators which are used to apply actions on parts such as a pusher.

Orienting parts with shape variation. In Chapter 2, we have focused on orienting a planar imperfect part with pushing by placing a single frictionless jaw in different orientations where the envelopes are convex polygonal objects. This is the first work which considers shape variation along the entire boundary of an object for parts orienting approach. We then have shown how the family of possible parts defined by this model can be simultaneously oriented into the smallest possible interval of final orientations after applying a bounded predetermined number of pushes denoted by $h > 0$. The computational time complexity of the proposed method is $O(hn)$. We have also considered the question of how many pushes are needed to reduce the interval of possible final orientations as much as possible. In many cases, additional pushes above some upper bound for h are superfluous in the sense that they cannot shrink the interval of possible orientations any further. However, we also showed that there are cases where one can only *approach* the optimal interval even with infinitly many pushes. We explored the conditions under which a finite number of pushes results in the smallest possible interval of

possible orientations. For the family of imperfect parts satisfying these conditions, exploring the relationship between the inner and outer envelopes and this upper bound on h is an interesting direction for future work.

We note that the final possible orientation set consists of disjoint intervals. Although our approach is able to compute this set of disjoint intervals, we did not exploit this information and focused on finding the smallest single interval that contains all these possible orientations. The location of the center of mass plays a key role in how a part settles upon being pushed. Our work assumes that though the shape is not known, center of mass is located at a given point. Our approach can be extended to the version in which the location of the center of mass can vary arbitrarily inside a disc to compute an approximation of the minimum set of final possible orientations. Another possible extension concerns the optimal solution in the case where the location of the center of mass is also allowed to vary.

Bounding the locus of the center of mass for parts with shape variation. In Chapter 3, we have studied the location of the center of mass for imperfect parts with uniform mass distribution under the same model. We have bounded the location of the center of mass for both two and three dimensional cases. We have identified the valid instance that maximizes the displacement of the center of mass in a given direction, and used this result to find a k-vertex polytopic outer approximation of the set of all possible center-of-mass loci in $O(kn)$ time, where n is the total number of the envelopes' vertices. Then, we turned our attention to the case that the outer envelope is within a distance of the inner envelope; this case is realistic, as the admitted shape variation for a manufactured part is usually small compared to the dimensions of the part itself. Then, we assumed that the inner envelope is both convex and β-fat by definition of fatness proposed by de Berg and van der Stappen [55]. We have shown that the diameter of the set of all possible locations of the center-of-mass is proportional to the maximum distance of the two envelopes and inversely proportional to β . It is interesting to see under which circumstances these results can be extended to parts with non-uniform mass distribution and also non-convex parts.

Pose statistics for eccentric parts. Pose statistics is a fundamental topic related to part orienting. In Chapter 4, we have studied the pose statistics problem for a family of 3D objects whose initial pose is uniformly random. Our approach can also be applied to parts in 2D and parts with shape variation. Since the goal of part orienting is to reduce the uncertainty in the orientation of a part, in many cases it is important to know the probability of resting in different poses when it is dropped on a surface. We assumed that the object falls onto a flat surface in presence of gravity under quasi-static conditions. We defined a type of geometric eccentricity that not only applies in any dimension but also distinguishes between different types of

object eccentricity. We focused on 3D eccentric objects which have two types of eccentricities. For a given $k > 1$, an eccentric 3D object is thin and long i.e. k times bigger in one dimension than the other dimensions or wide and flat i.e. k times bigger in two dimensions than the other dimension. For eccentric objects, we have shown that the object with high probability rests at a pose which is close to a specific plane or specific line.

These results show the relation between the part eccentricity and the maximum distance of the final pose to a unique plane or a line as well as its probability. We saw that for larger k, there is a smaller range of poses at which the eccentric objects end up with higher probability.

Our results are based on the quasi-static assumption. The work by Goldberg et al. [71] and Várkonyi [134] suggests that our results are therefore conservative because more realistic part behavior models tend to lead to even larger biases towards even smaller ranges of poses. To rigorously show this, it is interesting to explore a model incorporating the dynamics of collisions and friction.

Our analysis of pose statistics for eccentric parts departs from the assumption that the object at hand is known to be eccentric. It is therefore natural to consider how k can be computed for a given object P as a future work. In particular, to obtain the strongest possible bounds on the bias towards certain poses it is desirable to find the largest k for which P is eccentric.

6.2 Proximity probing

Part III of this thesis has investigated a novel type of geometric probing. Geometric probing is an area of research with various applications in robotics and automated manufacturing. The goal is to interactively determine geometric shape and orientation of an unknown objects by using special measurements. Existing approaches to shape measurement generally use sensors with high-dimensional output, such as cameras, or use low-dimensional sensors to measure fixed points, such as with Scanning Probe Microscopy (SPM). It is shown here that by employing an algorithm that uses the results of previous measurements to determine how the next measurement is taken, a non-directional range sensor can be used to efficiently and exactly determine the shape of a convex polygon. This suggests an alternative approach to obtaining information on the shape of an object in cases where low-dimensional sensors are more accurate, faster, or cheaper than their counterparts.

Proximity probing algorithms for metrology. In Chapter 5, we have defined a type of proximity probing which returns the distance to the boundary of the object in question. This work has concentrated on the case where the object is a convex polygon P in the plane which has n vertices, $k \leq 3$ of which are acute angle vertices. The goal is to find an upper bound on the number

of measurements required to exactly determine P. We have proposed an algorithm requiring at most $3.5n + k + 2$ probes. Furthermore, our method is computationally very efficient, requiring only constant computation time per probe, for a total time complexity of $O(n)$.

We also considered the same task of using these proximity probes to identify P, but from a finite set of convex polygons. In many applications, the object is unknown but the set of possible objects is known. Given a set of m convex polygons each of which has at most n' edges, we presented an algorithm achieving this using at most $2n + 2$ probes. The time complexity of each object query is linear in the size of the object and the prepossessing time is $O(n'm)$.

An interesting extension would be to find lower bounds for the number of probes needed by any algorithm, which would allow us to measure the performance of our algorithm (and any future algorithms) against a hypothetical optimal solution. Another important avenue for future work is to study the case where the probe measurements are not precise and instead lie within some known bounds of the true value. Instead of precisely identifying the unknown polygon, it is useful to give an approximation of it. This extension is particularly relevant as real measurements are never completely precise. Other possibilities include extending these results to higher dimensions and also to non-convex objects. Studying the problem for the non-convex objects is inspired by the approach that Boissonnat and Yvinec [33] developed to extend finger probes to non-convex polyhedra. Another version of the problem is using proximity probes from inside the polygon. It is also interesting to consider the use of parallel probes rather than sequential ones where we need a faster procedure for deducing the object of interest. In this version, at each step we specify multiple probes, rather than a single probe; this will decrease the number of steps but increase the total number of probes needed.

To summarize in a sentence, we have proposed geometric paradigms and algorithmic solutions for different tasks in the field of automation. I hope that this thesis motivates researchers to further explore the field so that the geometric designs are more reliable in practice.

Bibliography

[1] P. K. Agarwal, A. Collins, and J. Harer, Minimal trap design, In *IEEE International Conference on Robotics and Automation (ICRA)*, 3: pp. 2243-2248, 2001.

[2] P. K. Agarwal, R. -P. Berretty, and A. D. Collins, A near-quadratic algorithm for fence design. *Discrete and Computational Geometry* 33(3): pp. 463-481, 2005.

[3] S. Akella, W. Huang, K. M. Lynch, and M. T. Mason, Parts feeding on a conveyor with a one joint robot. *Algorithmica*, 26: pp. 313-344, 2000.

[4] S. Akella, and M. T. Mason. Posing polygonal objects in the plane by pushing. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2255-2262, 1992.

[5] S. Akella, and M. T. Mason, Orienting Toleranced Polygonal Parts. *International Journal of Robotics Research*, 19(12): pp. 1147-1170, 2000.

[6] E. M. Arkin, M. Held, J. Mitchell, and S. S. Skiena, Recognizing polygonal parts from width measurements. *Computational Geometry: Theory and Applications*, 9(4): pp. 237-246, 1998.

[7] V. Arnold, *Arnolds problems*. Springer, Berlin-Heidelberg-NewYork and PHASIS, Moscow, 2005.

[8] B. Aronov, and M. de Berg, Unions of fat convex polytopes have short skeletons. *Discrete and Computational Geometry*, 48(1): 53-64, 2012.

[9] ASME Y14.5, Dimensioning and Tolerancing Standard, *The American Society of Mechanical Engineers*, New York, 1994.

[10] ASME Y14.5M, Mathematical definition of dimensioning and tolerancing principles, *The American Society of Mechanical Engineers*, 1994.

[11] F. Barthe, The Brunn-Minkowski theorem and related geometric and functional inequalities. In *International Congress of Mathematicians*, 2: pp. 1529-1546, 2006.

[12] M. Berger, and B. Gostiaux, *Differential geometry: Manifolds, curves and surfaces.* Springer, New York, 1988.

[13] M. Bern, D. Eppstein, L. J. Guibas, J. E. Hershberger, S. Suri, and J. Wolter, The centroid of points with approximate weights. In *3rd European Symposium Algorithms*, pp. 460-472, 1995.

[14] H. J. Bernstein, Determining the shape of a convex n-sided polygon by using $2n + k$ Tactile Probes. *Information Processing Letters*, 22(5): pp. 255-260, 1986.

[15] R. -P. Berretty, *Geometric Design of Part Feeders.* PhD thesis, Utrecht University, Utrecht, The Netherlands, 2000.

[16] R. -P. Berretty, M. H. Overmars, and A. F. van der Stappen, Orienting polyhedral parts by pushing. *Computational Geometry: Theory and Applications* 21(1-2): pp. 21-38, 2002.

[17] R. -P. Berretty, K. Y. Goldberg, M. H. Overmars, and A. F. van der Stappen, Computing fence designs for orienting parts. *Computational Geometry: Theory and Applications*, 10(4): pp. 249-262, 1998.

[18] R. -P. Berretty, K. Y. Goldberg, M. H. Overmars, and A. F. van der Stappen, Orienting parts by inside-out pulling. In *IEEE International Conference on Robotics and Automation* pp. 1053-1058, 2001.

[19] R. -P. Berretty, K. Y. Goldberg, M. H. Overmars, and A. F. van der Stappen, Geometric algorithms for trap design. In *Annual ACM Symposium on Computational Geometry*, pp. 95-104, 1999.

[20] R. -P. Berretty, K. Y. Goldberg, M. H. Overmars, and A. F. van der Stappen, Geometric trap design for automatic part feeders. In *International Symposium of Robotics Research*, pp. 139-144, 1999.

[21] R. -P. Berretty, K. Y. Goldberg, M. H. Overmars and A. F. van der Stappen. Trap design for vibratory bowl feeders. *International Journal of Robotics Research*, 20(11): pp. 891-908, 2001.

[22] R. -P. Berretty, K. Y. Goldberg, M. H. Overmars, and A. F. van der Stappen, On Fence Design and the complexity of push plans for orienting parts. In *13th ACM Symposium on Computational Geometry*, pp. 21-29, 1997.

[23] K. -F. Böhringer, V. Bhatt, B. R. Donald, and K. Y. Goldberg, Algorithms for sensorless manipulation using a vibrating surface. *Algorithmica*, 26: pp. 389-429, 2000.

[24] K. -F. Böhringer, B. R. Donald, and N. C. MacDonald. Upper and lower bounds for programmable vector fields with applications to MEMS and vibratory plate parts feeders. *International Workshop on Algorithmic Foundations of Robotics*, pp. 255-276, 1996.

[25] K. -F. Böhringer, B. R. Donald, and N. C. MacDonald. Programmable vector fields for distributed manipulation, with applications to MEMS actuator arrays and vibratory parts feeders. *International Journal of Robotics Research*, 18(2): 168-200, 1999.

[26] K. -F. Böhringer, V. Bhatt, and K. Y. Goldberg, Sensorless manipulation using transverse vibrations of a plate. *IEEE International Conference on Robotics and Automation* pp. 1989-1996, 1995.

[27] K. -F. Böhringer, B. Donald, and N. Macdonald, Programmable force fields for distributed manipulation, with applications to MEMS actuator arrays and vibratory parts feeders. *International Journal of Robotics Research*, 18(2): 168-200, 1999.

[28] K. -F. Böhringer, B. Donald, R. Mihailovich, and N. Macdonald, Sensorless manipulation using massively parallel microfabricated actuator arrays. In *IEEE International Conference on Robotics and Automation* pp. 826-833, 1994.

[29] K. -F. Böhringer, B. Donald, L. Kavaraki, and F. Lamiraux, Part orientation with one or two stable equilibria using programmable force fields. *IEEE Transactions on Robotics and Automation*, 16(2): pp. 157-170, 2000.

[30] K. -F. Böhringer, and H. Choset, *Distributed manipulation.* Kluwer Academic Publishers, pp. 826-833, 2000.

[31] K. -F. Böhringer, B. R. Donald, L. E. Kavraki, and F. Lamiraux. A distributed, universal device for planar parts feeding: unique part orientation in programmable force fields. *Distributed Manipulation*, Kluwer Academic Publishers, pp. 1-28, 2000.

[32] K. -F. Böhringer, B. R. Donald, and N. C. MacDonald, What Programmable Vector Fields Can (and Cannot) Do: Force Field Algorithms for MEMS and Vibratory Plate Parts Feeders. *IEEE International Conference on Robotics and Automation*, 1: pp. 822-829, 1996.

[33] J. D. Boissonnat, and M. Yvinec, Probing a scene of non-convex polyhedra. *Algorithmica*, 8: pp. 321-342, 1992.

[34] S. J. Blind, C. C. McCullough, S. Akella, and J. Ponce, Manipulating Parts with an Array of Pins: A Method and a Machine. *International Journal of Robotics Research* 20(10): pp. 808-818, 2001.

[35] A. Bicchi, and R. Sorrentino, Dexterous manipulation through rolling. In *IEEE International Conference on Robotics and Automation*, 1: pp. 452-457, 1995.

[36] J. D. Boissonnat, D. Cohen-Steiner, B. Mourrain, G. Rote, and G. Vegter, *Effective Computational Geometry for Curves and Surfaces.* Springer-Verlag, 2006.

[37] G. Boothroyd, and P. Dewhurst. *Design for assembly - A designers handbook.* Department of Mechanical Engineering, University of Massachusetts, Amherst, Mass, 1983.

[38] G. Boothroyd, C. Poli, and L. Murch, *Automatic assembly.* Marcel Dekker, Inc., New York, 1982.

[39] G. Boothroyd, A. H. Redford, C. Poli, and L. E. Murch, Statistical distributions of natural resting aspects of parts for automatic handling. *Manufacturing Engineering Transactions*, 1: pp. 93-105, 1972.

[40] G. Boothroyd, and C. Ho, Natural resting aspects of parts for automatic handling. *Journal of Engineering for Industry*, 99: pp. 314-317, 1977.

[41] M. Brokowski, M. A. Peshkin, and K. Y. Goldberg, Optimal curved fences for part alignment on a belt. *ASME Transactions of Mechanical Design*, 117(1): pp. 27-35, 1995.

[42] R. C. Brost, Dynamic analysis of planar manipulation tasks. In *IEEE International Conference on Robotics and Automation*, Vol.3, pp. 2247-2254, 1992.

[43] R. C. Brost, A cad-based approach to general-purpose part feeders. *Industries Association Flexible Parts Feeding Workshop*, 1993.

[44] R. Brost, and R. Peters, Automatic design of 3D fixtures and assembly pallets. *IEEE Int. Conference on Robotics and Automation*, 1: pp. 495-502, 1996.

[45] B. R. Donald, *Error Detection and Recovery in Robotics*. Springer-Verlag, 1987.

[46] J. Chen, K. Y. Goldberg, M. H. Overmars, D. Halperin, K. -F. Böhringer, and Y. Zhuang, Shape tolerance in feeding and fixturing. *(P. K. Agarwal, L. E. Kavraki, and M. T. Mason Eds.) Robotics: the algorithmic perspective*, pp. 297–311. A.K. Peters, 1998.

[47] J. Canny, and K. Y. Goldberg, A RISC approach to robotics. *IEEE Robotics and Automation Society*, 1(1): pp. 26-28, 1994.

[48] J. Canny, and K. Y. Goldberg, RISC for industrial robotics: Recent results and open problems. In *IEEE International Conference on Robotics and Automation*, pp. 1951-1958, 1994.

[49] J. Chen, K. Y. Goldberg, M. H. Overmars, D. Halperin, K. Böhringer, and J. Zhuang, Computing tolerance parameters for fixturing and feeding. *Assembly Automation*, 22(2): 163-172, 2002.

[50] Y. -B. Chen, and D. J. Ierardi, The complexity of oblivious plans for orienting and distinguishing polygonal parts. *Algorithmica*, 14: pp. 367-397, 1995.

[51] P. S. K. Chua, and M. L. Tay, Modelling the natural resting aspect of small regular shaped parts. *Manufacturing Science and Engineering*, 120: pp. 540-546, 1998.

[52] R. Cole, and C. K. Yap, Shape from probing. *Journal of Algorithms*, 8(1): pp. 19-38, 1987.

[53] H. Czichos, T. Saito, and L. E. Smith, *Handbook of Metrology and Testing*, 2011.

[54] M. Davoodi, and A. Mohades, Data imprecision under λ-geometry: Range searching problems. *Scientia Iranica*, 20(3): pp. 663-669, 2013.

[55] M. de Berg, and A. F. van der Stappen, On the fatness of Minkowski sums. *Information Processing Letter*, 81(5), pp. 259-264, 2001.

[56] D. Dobkin, H. Edelsbrunner, and C. K. Yap, Probing convex polytopes. In *18th Annual ACM Symposium on Theory of Computing*, pp. 424-432, 1986.

[57] M. Dogar, and S. S. Srinivasa, A framework for push-grasping in clutter. *Robotics: Science and Systems* 2, 2011.

[58] G. Domokos, J. Papadopoulos, and A. Ruina, Static equilibria of planar, rigid bodies: is there anything new? *J. Elasticity*, 36: pp. 59-66, 1994.

[59] G. Domokos, A. Sipos, T. Szabó, and P. Várkonyi, Pebbles, shapes, and equilibria, *Math Geosci* 42: pp. 29-47, 2010.

[60] C. L. Dotson, *Fundamentals of Dimensional Metrology*, 2006.

[61] V. Eichhorn, M. Bartenwerfer, and S. Fatikow, Nanorobotic assembly and focused ion beam processing of nanotube-enhanced AFM probes, *IEEE Transactions on Automation Science and Engineering*, 9(4): pp. 679-686, 2012.

[62] M. A. Erdmann, and M. T. Mason, An exploration of sensorless manipulation. *IEEE Journal of Robotics and Automation*, 4: pp. 367-379, 1988.

[63] M. A. Erdmann, M. Mason, and G. Vanecek. Mechanical parts orienting: The case of a polyhedron on a table. *Algorithmica*, 10: 226-247, 1993.

[64] P. Frei, *Theory, Design and Implementation of a novel vibratory conveyor.* PhD thesis, Swiss Federal Institute of Technology, 2002.

[65] Y. Giora, and H. Kaplan Optimal dynamic vertical ray shooting in rectilinear planar subdivisions. In *18th ACM-SIAM Symposium on Discrete Algorithms*, pp. 19-28, 2007.

[66] O. C. Goemans, M. T. Anderson, K. Y. Goldberg, and A. F. van der Stappen, Automated feeding of industrial parts with modular blades: design software, physical experiments, and an improved algorithm. In *IEEE Conference on Automation Science and Engineering*, pp. 318-325, 2007.

[67] O. C. Goemans, A. Levandowski, K. Y. Goldberg, and A. F. van der Stappen, On the design of guillotine traps for vibratory bowl feeders, In *IEEE Conference on Automation Science and Engineering*, pp. 79-86, 2005.

[68] O. C. Goemans, K. Y. Goldberg, and A. F. van der Stappen, Blades: a new class of geometric primitives for feeding 3D parts on vibratory tracks. In *IEEE International Conference on Robotics and Automation*, pp. 1730-1736, 2006.

[69] O. C. Goemans, and A. F. van der Stappen, On the design of traps for feeding 3d parts on vibratory tracks. In *IEEE International Conference on Robotics and Automation*, pp. 385-392, 2008.

[70] K. Y. Goldberg. Orienting polygonal parts without sensors. *Algorithmica*, 10(2): pp. 201-225, 1993.

[71] K. Y. Goldberg, B. Birtich, Y. Zhuang, J. Craig, B. Carlisle, and J. Canny. Part pose statistics: estimators and experiments. *IEEE Transactions on Robotics and Automation*. 15(5): pp. 849-857, 1999.

[72] C. M. Gray, *Algorithms for Fat Objects: Decompositions and Applications.* PhD thesis Technische Universiteit Eindhoven, Netherlands, 2008.

[73] D. Gudmundsson, and K. Y. Goldberg, Tuning robotic part feeder parameters to maximize throughput. *Assembly Automation*, 19(3): pp. 216-221, 1999.

[74] S. Guha, and K. T. Khańh, Recognizing convex polygons with few finger probes. *Pattern Analysis and Applications*, 12(2): pp. 193-199, 2009.

[75] L. Guibas, D. Salesin, and J. Stolfi, Epsilon geometry: Building robust algorithms for imprecise computations. In *5th ACM Annual Symposium on Computational Geometry*, pp. 208-217, 1989.

[76] S. Gupta, and J. U. Turner, Variational solid modelling for tolerance analysis. *IEEE Computer Graphics and Applications*, 13(3): pp. 64-74, 1993.

[77] P. C. Hammer, The centroid of a convex body. In *Amer. Math. Soc.*, 2: pp. 522-525, 1951.

[78] M. Inui, M. Miura, and F. Kimura, Positioning conditions of parts with tolerances in an assembly. In *IEEE International Conference on Robotics and Automation*, 3: pp. 2202-2207, 1996.

[79] Y. -B. Jia, and M. Erdmann, The complexity of sensing by point sampling. In *First Workshop on the Algorithmic Foundations of Robotics*, pp. 283-300, 1994.

[80] E. Joseph, and S. S. Skiena, Model-based probing strategies for convex polygons. *Computational Geometry: Theory and Applications*, 2: pp. 209-221, 1992.

[81] L. Joskowicz, Y. Ostrovsky-Berman, and Y. Myers, Efficient representation and computation of geometric uncertainty: The linear parametric model. *Precision Engineering*, 34(1): pp. 2-6, 2010.

[82] S. V. Kalinin, and A. Gruverman, *Scanning Probe Microscopy of Functional Materials*, 2011.

[83] M. J. Katz, M. H. Overmars, and M. Sharir, Efficient hidden surface removal for objects with small union size. *Computational Geometry: Theory and Applications*, 2: pp. 223-234, 1992.

[84] L. E. Kavraki, Part orientation with programmable vector fields: Two stable equilibria for most parts. In *IEEE International Conference on Robotics and Automation*, 3: pp. 2446-2451, 1997.

[85] B. Kehoe, D. Berenson, and K. Y. Goldberg, Toward cloud-based grasping with uncertainty in shape: estimating lower bounds on achieving force closure with zero-slip push grasps. In *IEEE International Conference on Robotics and Automation*, 2012.

[86] K. Klein, and S. Suri, Capture bounds for visibility-based pursuit evasion. In *29th ACM Annual Symposium on Computational Geometry*, pp. 329-338, 2013.

[87] D. J. Kriegman, Let them fall where they may: Capture regions of curved objects and polyhedra. *International Journal of Robotics Research*, 16(4): pp. 448-472, 1997.

[88] F. Lamiraux, and L. Kavraki, Positioning of symmetric and non-symmetric parts using radial and constant fields: Computation of all equilibrium configurations. *International Journal of Robotics Research*, 20(8): pp. 635-659, 2001.

[89] J. -C. Latombe, R. Wilson, and F. Cazals, Assembly sequencing with toleranced parts, *Computer-Aided Design* 29(2): pp. 159-174, 1997.

[90] S. M. Lavalle, Planning algorithms, Chapter 12: Planning under sensing uncertainty. *Cambridge University press*, 2006.

[91] S. S. G. Lee, B. K. A. Ngoi, L. E. N. Lim, and S. W. Lye, Determining the probabilities of natural resting aspects of parts from their geometries. *Assembly Autom*, 17: pp. 137-142, 1997.

[92] S. -Y. R. Li, Reconstruction of polygons from projections. *Information Processing Letters*, 28: pp. 235-240, 1988.

[93] S. Lien, and J. T. Kajiya, A symbolic method for calculating the integral properties of arbitrary non-convex polyhedra. *IEEE Transactions on Computer Graphics and Applications*, 4: pp. 35-41, 1984.

[94] M. Lindenbaum, and A. Bruckstein, Blind Approximation of Planar Convex Sets. *IEEE Transactions on Robotics and Automation*, 10(4): pp. 517-529, 1994.

[95] M. Löffler, *Data Imprecision in Computational Geometry.* PhD thesis, Utrecht University, 2009.

[96] M. Löffler, and J. Phillips. shape fitting on point sets with probability distributions. In *17th European Symposium on Algorithms* LNCS 5757: pp. 313-324, 2009.

[97] T. Lozano-Perez, Motion planning and the design of orienting devices for vibratory part feeders. *MIT AI Lab Technical Report*, 1986.

[98] T. Lozano-Perez, M. T. Mason, and R. H. Taylor, Automatic synthesis of fine-motion strategies for robots. *International Journal of Robotics Research* 3(1): pp. 3-24, 1984.

[99] K. M. Lynch, and M. T. Mason., Stable pushing: Mechanics, controllability, and planning. *International Journal of Robotics Research*, 15(6): pp.533-556, 1996.

[100] K. M. Lynch, M. Northrop, and P. Pan, Stable limit sets in a dynamic parts feeder. *IEEE Transactions on Robotics and Automation*, 18(4): pp. 608-615, August 2002.

[101] M. T. Mason, *Manipulator grasping and pushing operations.* PhD thesis, MIT, 1982, Published in Robot Hands and the Mechanics of Manipulation, MIT Press, Cambridge, 1985.

[102] H. Meijer, and S. S. Skiena, Reconstructing polygons from X-Rays. *Geometriae Dedicata*, 61(2): pp. 191-204, 1996.

[103] B. Mirtich, Y. Zhuang, K. Y. Goldberg, J. Craig, R. Zanutta, B. Carlisle, and J. F. Canny: Estimating pose statistics for robotic part feeders. In *IEEE International Conference on Robotics and Automation*, pp. 1140-1146, 1996.

[104] B. Mokaberi, and A. A. G. Requicha, Drift compensation for automatic nanomanipulation with scanning probe microscopes, *IEEE Transactions on Automation Science and Engineering*, 3(3): pp.199-207, 2006.

[105] M. Moll, and M. A. Erdmann, Manipulation of pose distributions. *Robotics Research*, 21(3): pp. 277-292, 2002.

[106] M. Moll, and M. A. Erdmann, Uncertainty reduction using dynamics. In *IEEE International Conference on Robotics and Automation*, pp. 3673-3680, 2000.

[107] J. -M. Morvan, *Generalized Curvatures*, (Part VI: The Steiner Formula). Springer Series in Geometry and Computing, 2: pp. 6-7, 2008.

[108] T. D. Murphey, J. Bernheisel, D. Choi, and K. M. Lynch, An example of parts handling and self-assembly using stable limit sets. In *IEEE/RSJ International Conference on Intelligent Robots and Systems.* pp. 1624-1629, 2005.

[109] B. K. Natarajan, Some paradigms for the automated design of parts feeders. *International Journal of Robotics Research*, 8(6): pp. 89-109, 1989.

[110] B. K. A. Ngoi, S. S. G. Lee, and L. E. N. Lim, Analysing the probabilities of the natural resting aspects of a component with a displaced centre of gravity. *Production Research*, 33(9): pp. 2387-2394, 1995.

[111] B. K. A. Ngoi, L. E. N. Lim, and S. S. G. Lee, Analyzing the probabilities of natural resting for a component with a virtual resting face. *Manufacturing Science and Engineering*, 120: pp. 468-471, 1998.

[112] B. K. A. Ngoi, and L. E. N. Lim, Analysing the natural resting aspect of a component for automated assembly using the energy envelope method. *Advanced Manufacturing Technology*, 12: 132-136, 1996.

[113] J. Niemann, Electrical measurements on nanoscale materials. *Keithley Instruments tutorial paper*, 2004.

[114] Y. Ostrovsky-Berman, and L. Joskowicz, Tolerance envelopes of planar mechanical parts with parametric tolerances. *Computer-Aided Design*, pp. 531-534, 2005.

[115] M. H. Overmars, and A. F. van der Stappen, Range searching and point location among fat objects. *Journal of Algorithms*, pp. 626-656, 1996.

[116] S. Pampuri, A. Schirru, G. A. Susto, C. De Luca, A. Beghi, and G. De Nicolao, Multistep virtual metrology approaches for semiconductor manufacturing processes. In *IEEE Conference on Automation Science and Engineering*, pp. 91-96, 2012.

[117] M. A. Peshkin, and A. C. Sanderson, Planning robotic manipulation strategies for workpieces that slide. *IEEE Journal of Robotics and Automation*, 4(5): pp. 696-701, 1988.

[118] M. A. Peshkin, and A. C. Sanderson, The motion of a pushed sliding workpiece. *IEEE Journal of Robotics and Automation*, 4(6): pp. 569-598, 1988.

[119] A. S. Rao, and K. Y. Goldberg, Manipulating algebraic parts in the plane. In *IEEE Transactions on Robotics and Automation*, 11(4): pp. 598-602, 1995.

[120] A. S. Rao, and K. Y. Goldberg, Shape from diameter: Recognizing polygonal parts with a parallel-jaw gripper. *International Journal of Robotics Research*, 13(1): pp. 16-37, 1994.

[121] A. A. G. Requicha, Toward a theory of geometric tolerancing. *International Journal of Robotics Research* 2, 1983.

[122] T. J. Richardson, Approximation of planar convex sets from hyperplane probes. *Discrete and Computational Geometry*, 18(2): pp. 151-177, 1997.

[123] U. Roy, and B. Li, Representation and interpretation of geometric tolerances for polyhedral objects. II.: Size, orientation and position tolerances. *Computer Aided Design*, 31(4): pp. 273-285, 1999.

[124] U. Roy, C. Liu, and T. Woo, Review of dimensioning and tolerancing. *Computer-Aided Design*, 23(7): pp. 466-483, 1991.

[125] S. S. Skiena, Interactive Reconstruction via Geometric Probing, In *IEEE*, 80(9): pp. 1364-1383, 1992.

[126] S. S. Skiena, Problems in geometric probing, *Algorithmica*, 4(4): pp. 599-605, 1989.

[127] S. S. Skiena, Probing convex polygons with half-planes. *Journal of Algorithms* 12: pp. 359-374, 1991.

[128] M. Suresha, K. A. Jagadeeshb, and P. Ashoka Varthanana, Determining the natural resting orientation of a part using drop test and theoretical methods. *Journal of Manufacturing Systems*, 32(1): pp. 220-227, 2013.

[129] G. A. Susto, A. Schirru, S. Pampuri, G. De Nicolao, and A. Beghi, An Information-Theory and Virtual Metrology-based approach to Run-to-Run Semiconductor Manufacturing Control, In *IEEE Conference on Automation Science and Engineering*, pp. 358-363, 2012.

[130] P. J. Swanson, R. R. Burridge, and D. E. Koditschek, Global asymptotic stability of a passive juggling strategy: A possible parts-feeding method. *Mathematical problems in Engineering* 1(3): pp. 193-224, 1995.

[131] A. F. van der Stappen, Motion Planning amidst Fat Obstacles. *Ph.D. Thesis, Utrecht University*, 1994.

[132] A. F. van der Stappen, R. -P. Berretty, K. Y. Goldberg, and M. H. Overmars, Geometry and Part Feeding. *Sensor-Based Intelligent Robots* (G.D. Hager, H.I. Christensen, H. Bunke, R. Klein Eds.), LNCS 2238, Springer Verlag, Berlin, pp. 259-281, 2002.

[133] A. F. van der Stappen, K. Y. Goldberg, and M. H. Overmars, Geometric eccentricity and the complexity of manipulation plans, *Algorithmica* 26(3/4): pp. 494-514, 2000.

[134] P. L. Várkonyi, Estimating part pose statistics with application to industrial parts feeding and shape design: new metrics, algorithms, simulation experiments and datasets. *IEEE Transactions on Robotics and Automation*, 11: pp. 658-667, 2014.

[135] P. L. Várkonyi, and G. Domokos, Mono-monostatic bodies: the answer to Arnold's question. *Mathematical Intelligencer*, 28(4): pp. 34-38, 2006.

[136] P. L. Várkonyi, and G. Domokos, Static equilibria of rigid bodies: dice, pebbles and the Poincaré-Hopf Theorem. *J. Non-linear Science*, 16: pp. 251-283, 2006.

[137] H. Voelker, A current perspective on tolerancing and metrology. *Manufacturing Review*, 6(4): 1993.

[138] Y. Wang, and M. Mason, Two-dimensional rigid-body collisions with friction. *ASME Journal of applied Mechanics*, 59: pp. 635-642, 1992.

[139] J. A. Wiegley, K. Y. Goldberg, M. Peshkin, and M. Brokowski, A complete algorithm for designing passive fences to orient parts. *Assembly Automation*, 17(2): pp. 129-136, 1997.

[140] J. Wiegley, A. Rao, and K. Y. Goldberg, Computing a statistical distribution of stable poses for a polyhedron. In *Allerton Conference Communications, Control and Computing*, 1992.

[141] M. T. Zhang, K. Y. Goldberg, G. Smith, R. -P. Berretty, and M. H. Overmars, Pin design for part feeding. *Robotica* 19(6): pp. 695-702, 2001.

[142] N. B. Zumel, *A non-prehensile Method for Reliable Parts Orienting.* PhD thesis, Carnegie Mellon University, 1997.

[143] N. B. Zumel, and M. Erdmann, Non-prehensile manipulation for orienting parts in the plan. In *IEEE International Conference on Robotics and Automation* pp. 3317-3323, 1997.

Samenvatting

In de robotica en automatisering komen veel taken voor die kunnen worden opgelost met behulp van technieken uit de computationele geometrie. In dit proefschrift hebben we laten zien hoe computationele geometrie kan worden toegepast op verscheidene onderzoeksproblemen in dit gebied. Onze nadruk lag op twee onderwerpen: oriënteren (*orienting*) en aftasten (*probing*). Het doel van oriënteren is om een industrieel onderdeel zo te draaien dat het een unieke gewenste oriëntatie bereikt, zonder dat de initiële oriëntatie van het onderdeel van tevoren bekend is. Het doel van aftasten is om de geometrie en oriëntatie van een onbekend object zo goed mogelijk te meten.

We hebben ons gericht op drie problemen op het gebied van oriënteren:

- Eerst hebben we een algemeen model ontwikkeld voor variaties in vormen om de onvolmaaktheid van geproduceerde onderdelen te modelleren. Ons model staat variaties toe op de gehele rand van een onderdeel. De mogelijke variaties zijn begrensd zijn door twee gegeven gesloten vormen: een binnenste en een buitenste grens.

  Ons model is toepasbaar op zowel tweedimensionale als driedimensionale objecten. Vervolgens hebben we gekeken naar het oriënteren van een tweedimensionaal onvolmaakt onderdeel door het vanuit verschillende oriëntaties te duwen met een balk zonder wrijving, onder de aanname dat de twee grenzen convexe polygonen zijn.

  We hebben laten zien hoe alle mogelijke variaties van dit onderdeel in een zo klein mogelijk interval van eindoriëntaties kan worden gebracht met een vooraf bepaald aantal duwbewegingen. Ook hebben we bestudeerd hoeveel bewegingen nodig zijn om dit interval zo klein mogelijk te maken. Verder hebben we gevallen aangetoond waarin het kleinst mogelijke interval niet kan worden bereikt met een eindig aantal bewegingen. In deze gevallen kan het optimale interval zelfs met willekeurig veel bewegingen hooguit benaderd worden. We kunnen wel van elke instantie berekenen wat het optimale interval is.

- Met behulp van hetzelfde variatiemodel hebben we onderzocht waar het zwaartepunt van een onvolmaakt onderdeel kan liggen. We hebben de verzameling van alle mogelijke zwaartepunten begrensd voor zowel twee- als driedimensionale onderdelen met een uniforme massaverdeling. We kunnen de instantie van een onderdeel vinden waarmee het zwaartepunt maximaal wordt verplaatst in een gegeven richting. Hiermee kunnen we de verzameling zwaartepunten van buiten benaderen met een polytoop van k hoekpunten in $O(kn)$ tijd, waarin n het totale aantal hoekpunten is op de binnengrens en buitengrens van het onvolmaakte onderdeel. Vervolgens hebben we ons gericht op gevallen waarin de buitengrens een beperkte afstand tot de binnengrens heeft. We hebben aangenomen dat de binnengrens convex en β-*fat* is volgens de definitie van *fatness* gegeven door De Berg en Van der Stappen [55].

  We hebben bewezen dat de diameter van de verzameling mogelijke zwaartepunten proportioneel is aan de maximale afstand tussen de binnen- en buitengrens, en omgekeerd proportioneel aan β.

- We hebben onderzoek gedaan naar *pose statistics*, een fundamenteel begrip voor de oriëntatie van onderdelen. Om de onzekerheid over de oriëntatie van een onderdeel te verminderen is het vaak belangrijk om te weten met welke kansen het onderdeel in bepaalde orieñtaties kan belanden wanneer het op een oppervlak valt. We hebben ons gericht op 3D-onderdelen waarvan de initiële orieñtatie uniform willekeurig verdeeld is. Onze aanpak is ook van toepassing op 2D-onderdelen en onderdelen met vormvariatie. We hebben aangenomen dat het onderdeel op een plat oppervlak valt onder invloed van zwaartekracht onder quasi-statische aannamen.

  Eerst hebben we een type geometrische excentriciteit (*eccentricity*) gedefinieerd die toepasbaar is in elke dimensie en waarvan verschillende soorten bestaan.

  Onze focus lag op 3D-objecten met twee soorten excentriciteit: objecten die in een dimensie veel groter zijn dan in de andere twee (en dus dun en lang zijn) en objecten die in twee dimensies veel groter zijn dan in de derde (en dus breed en plat zijn). Voor deze objecten hebben we aangetoond dat ze met grote waarschijnlijkheid in een orieñtatie belanden die dichtbij een specifiek vlak of specifieke lijn ligt.

  Deze resultaten tonen de relatie aan tussen excentriciteit en de maximale afstand van de uiteindelijke orieñtatie tot een uniek vlak of unieke lijn, evenals de kans waarmee dit vlak of deze lijn benaderd wordt. Hoe groter de excentriciteit, hoe groter de waarschijnlijkheid dat het object terechtkomt in een kleine verzameling orieñtaties.

Het tweede onderwerp van dit proefschrift is aftasten, een onderzoeksgebied met veel toepassingen in robotica en automatische productie. Bij aftasten proberen we de vorm en orieëntatie van een onbekend onderdeel interactief te bepalen met speciale meettechnieken.

- We hebben een type aftasting (*proximity probing*) gedefinieerd dat de afstand tot de rand van een onderdeel kan bepalen. Hierbij richtten we ons op het geval waarin het onderdeel een convexe veelhoek P is in het platte vlak, met n hoekpunten waarvan er $k \leq 3$ een scherpe hoek maken. Het doel is om een bovengrens te vinden op het aantal afstandsmetingen (*probes*) om de vorm van P exact mee te bepalen. We hebben een algoritme ontwikkeld dat hooguit $3.5n + k + 2$ metingen gebruikt. Bovendien gebruikt onze meetmethode slechts constante tijd per meting, waardoor de totale looptijd van het algoritme $O(n)$ is.
- Daarnaast hebben we dezelfde taak bestudeerd waarbij P een convexe veelhoek is uit een eindige verzameling bekende convexe veelhoeken. In veel toepassingen is het onderdeel zelf onbekend, maar zijn alle mogelijkheden wel bekend. Voor een verzameling van m convexe veelhoeken hebben we een algoritme ontworpen dat met hooguit $2n + 2$ metingen bepaalt welk element P is. De querytijd per onderdeel in de verzameling is lineair in het aantal hoekpunten van het onderdeel.

Curriculum Vitae

Fatemeh Panahi was born on 19 February 1985 in Zanjan, Iran. She finished her pre-university education in 2003 in NODET (National Organization for Development of Exceptional Talents), Zanjan. In 2007 she received her Bachelor degree in Computer Science from Shahid Beheshti University of Technology, Tehran. She then continued with a Master in Computer Science, at Amirkabir University of Technology (Polytechnic), Tehran. Her Master thesis was on Geometric Algorithms for Point Matching, under the supervision of dr. Ali Mohades. She received her Master degree in December 2009. Afterward, she worked as a research and development specialist in Iran-tranfso company in Tehran. Since October 2011 she has been working as a PhD Student in the Virtual Words group at Utrecht University. The results of her PhD are presented in this thesis.

Other Publications

In addition to the publications incorporated into this thesis, Fatemeh has published the following formally reviewed articles.

- *M. Davoodi, F. Panahi, A. Mohades, S. N. Hashemi, Clear and Smooth Path Planning, Applied Soft Computing, 32, pp. 568-579, 2015.*
- *M. Davoodi, F. Panahi, A. Mohades, S. N. Hashemi, Multi-Objective Path Planning in Discrete Space, Applied Soft Computing13(1): pp. 709-720, 2013.*
- *A. Gheibi, M. Davoodi, A. Javad, F. Panahi, M. M. Aghdam, M. Asgaripour, A. Mohades, Polygonal Shape Reconstruction in the Plane, IET Computer Vision, 5(2): pp. 97-106, 2011.*
- *F. Panahi, A. Mohades, M. Davoodi, Weak Matching Points Using Triangles, In Canadian Conference on Computational Geometry, 2011.*
- *M. Davoodi, F. Panahi, A. Mohadesi, Clear and Smooth path planning, In Contemporary Issues in Computer and Information Sciences, 2011.*

- *F. Panahi and A. Mohades, Computing Minimum Limited-Capacity Matching for Two Point Sets on the Real Line and on Tow Perpendicular Lines, In Canadian Conference on Computational Geometry, 2010.*
- *F. Panahi , A. Mohades, Minimum-Cost Limited-Capacity Many-To-Many Point Matching, In 7th Japan Conference on Computational Geometry and Graphs, 2009.*

Acknowledgments

It is a pleasure to think back at these past four years and remember anyone who directly or indirectly, has contributed to the progress and publication of the work contained herein. I owe my gratitude to all these people because of whom my graduate experience has been a great joy that I will cherish forever.

In the first place, I would like to deeply thank my supervisor Frank van der Stappen for giving me the opportunity to pursue my PhD under his supervision in the Virtual Worlds group in Utrecht University. I can hardly express my gratitude in words to Frank. I appreciate all his contributions of time, knowledge and ideas to make my PhD experience productive and stimulating. The combination of his thorough skepticism of every thought I presented him with a strong level of his support, patience and encouragement improved me greatly as a researcher and a person. He has been overly generous with his time and his experience. Whenever I got stuck in a research problem, I could easily walk into his office and he was always open to give me fruitful guidelines to proceed. Thank you Frank for being such a supportive, friendly and considerate supervisor. I will definitely miss our weekly research meetings and also our discussions about other interesting topics.

My gratitude is also due to Marc van Kreveld. I benefited immensely from his assistance in finalizing my thesis. I also thank him for the concepts I learned during the courses that I took with him or on which I was his assistant. I thank him and also Mark Overmars for the great support of the Virtual Worlds group.

I am deeply indebted to the members of the reading committee: Hans Bodlaender, Leo Dorst, Ken Goldberg, Rolf Klein, and Sandor Fekete for reviewing my thesis and for their useful remarks. Most of the work in this thesis appeared in the proceedings of various conferences or has been published in journals. I would like to thank all anonymous referees who have read and checked most parts of the thesis.

It was a great privilege for me to visit Ken Goldberg's group at UC Berkeley during my PhD. I thank Ken and his students for their pleasant hospitality

during my visit. The visit and continued collaboration with the group afterwards has led to the results appearing in Chapter 5 of this thesis. I would like to thank Aviv Adler, with whom I started working on a project when we were both visiting UC Berkeley; he became my coauthor afterward; and now he is one of my valuable friends. I have had many discussions with him when we visited in conferences and workshops, during his visit of Utrecht University, and also during our long Skype conversations not only about computer science, but also many other interesting topics such as philosophy, cultures, religions, music, life, etc. His sense of humor and his impressive motivation have been an inspiration and a pleasure for me. He has also been my endless source of cool and weird English words. My thanks also go to him for all his efforts in reading and correcting some pages of my thesis. I also indebted my other coauthor Mansoor Davoodi. The joy, enthusiasm and perseverance he has for doing research is always contagious and motivational for me.

Of course, the path to my doctorate started before Utrecht University. I would like to thank Ali Mohades, my supervisor for my masters degree at the Algorithms and Computational Geometry (ACG) group of Amirkabir University of Technology. I took his interesting course on computational geometry and it was where I discovered the great joy and fun in this field. He is responsible for making me so enthusiastic to do research; Mark de Berg, Günter Rote, Rolf Klein and Elmar Langetepe share the blame as well for their impressive talks during the First and Second Winter Schools on Computational Geometry in Amirkabir University of Tehran.

My time in Utrecht was made enjoyable in large part due to cherished friends who provided enthusiasm and empathy in just the right moments. I would like to thank all the members of Virtual Worlds group for all the good times I spent with them during the coffee breaks, lunch time and social get-togethers. Particularly Frank Staals who was always helpful, patient and friendly toward my questions and Wouter van Toll for being my go-to person for any questions about my Dutch letters or other translations. I am also very thankful to Rita Jansen and Corine Kranenberg-Jolles for being kindly insistent to help whenever I asked. During these past four years, I shared my office with Heinrich Kruger, Cathy Ennis, Mart Hagenaars, Zümra Kavafoğhlu and Jerome Urhausen. I have had a great time with all of them and it was a good fortune to have such nice officemates, each from a different country that gave me the opportunity to learn something from each. Cathy and Zümra acted as kind angels during their short stay in Utrecht. Zümra - the always smiling buddy - has brought a new pulse of energy to the office. We had a great time thanks to all great discussions, concerns and laughs that we shared together. I thank Zerrin Yumak and Yasaman Fereidooni for agreeing to support me during the defense ceremony as paranymphs.

Thanks to the companionship of all my friends particularly Mostafa Va-

hedi, Maryam Faridounnia and Amir Saeadi which helped me to quickly adapt myself to my new life during the first month of living in Utrecht. My gratitude is due to the wonderful couple Wilke Schram and his wife Tineke. I never forget the help and the great times I spent with them. I would like to thank Maryam with whom I have spent long hours studying at Utrecht library. She was a big motivation for me to spend more time to read and she was very generous to lend me her valuable books. I thank my dear friends Maryam, Mojtaba, Yasaman, Li, Jonas, Petra, Forough and many others with whom I have spent many pleasurable times in Utrecht. I also thank my kind neighbor Shahnaz and her family for their great support, I always felt home here and I have always felt they are my second family in Utrecht.

I would have never reached the place where I am now without the endless love, encouragement and support of my lovely parents. I am immensely thankful for their trust and confidence in me, which has allowed me to discover many aspects of life, independently. Many thanks to my dear brothers Esmaeel and Ebrahim, and my lovely sisters Nahid, Zahra, Leyla and Elaheh for all their help and back up and the nice moments we have spent together.

Last, but far from least, I would like to deeply thank my dear partner Afshar for being an endless source of energy, love, and encouragement despite our huge physical distance. I met Afshar ten years ago when he was our teaching assistant during my undergraduate study. I knew him as the smartest student in our department. Our new life as a couple started during my PhD and he is still the most intelligent, organized, honest and loyal guy I have ever met. I experienced the wonderful feeling of discovering a great portion of the world with him. I cherish our amazing trips all over the world, biking, patinaje and hiking experiences and many other invaluable moments. Infinite thanks to you for being my better half and my dearest companion.