

Clamshell Casting!

By
Stefanie Wuhler

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfilment of
the requirements for the degree of
Master of Computer Science

Ottawa-Carleton Institute for Computer Science
School of Computer Science
Carleton University
Ottawa, Ontario

June 2006

© Copyright
2006, Stefanie Wuhler

The undersigned hereby recommend to
the Faculty of Graduate Studies and Research
acceptance of the thesis,

Clamshell Casting!

submitted by

Stefanie Wuhrer

Douglas Howe
(Director, School of Computer Science)

Prosenjit Bose
(Thesis Supervisor)

Pat Morin
(Thesis Supervisor)

Michiel Smid
(Thesis Supervisor)

Carleton University

June 2006

Abstract

A popular manufacturing technique is casting, where liquid is poured into a cast and the cast is removed once the liquid has hardened. We consider *clamshell casting* in two and three dimensions where the cast consists of two parts and is removed by rotation from the boundary of the object to be manufactured.

In two and three dimensions, the cast is modeled by a simple polygon and a polyhedron, respectively. Given a cast and a center or axis of rotation, we determine whether there exists a partitioning of the cast into two parts, such that both parts can be removed by opposite rotations without colliding with the interior of the object to be manufactured or the cast. Furthermore, an algorithm is presented to find all the centers or axes of rotation that allow a cast partitioning as described above.

Acknowledgments

I would like to thank my supervisors Prosenjit Bose, Pat Morin, and Michiel Smid for their continual advice and various support. Being experts in the field of computational geometry, my supervisors shared their knowledge and experience and always pointed to relevant literature.

I wish to thank Anil Maheshwari for sharing his insight in the field of algorithms with me.

Furthermore, I thank all of the members of the Computational Geometry Group at Carleton University, the School of Computer Science at Carleton University, Graduate Studies at Carleton University, and the Ontario Student Assistantship Program for their generous financial support.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Problem Statement	3
1.3	Related Work	4
1.4	Summary of Results and Thesis Organization	9
2	Rotational Casting in Two Dimensions	11
2.1	Preliminaries	11
2.2	Decision Problem	16
2.3	Determining all valid regions for cast removal	19
2.3.1	Convex polygons	19
2.3.2	Simple polygons	26
3	Rotational Casting in Three Dimensions	34
3.1	Preliminaries	34
3.2	Decision Problem	37
3.2.1	Robust rotational casting	37
3.2.2	General rotational casting	45
3.3	Determining all valid casting lines	52
3.3.1	Robust rotational casting	53
3.3.2	General rotational casting	66
4	Conclusion and Future Work	77

List of Tables

1.1	<i>Summary of the results for two-dimensional clamshell casting.</i>	10
1.2	<i>Summary of the results for three-dimensional clamshell casting.</i>	10

List of Figures

1.1	<i>Figure (a): Copper frog cast around 3200 B.C. Figure (b): Bronze elephant cast around 1122 to 255 B.C. Figures out of Simpson's book [37].</i>	2
1.2	<i>Figure (a): The polygon P is partitioned into exactly two connected chains, such that the solid chain can be rotated clockwise around r and the dashed chain can be rotated counterclockwise around r without intersecting the interior of P. Figure (b): The polyhedron P is partitioned into exactly two connected parts, such that the shaded part can be rotated clockwise around l and the white part can be rotated counterclockwise around l without intersecting the interior of P. . . .</i>	4
2.1	<i>The edges e_1 and e_2 are removable in cw orientation with angle α and ccw orientation with angle β with respect to r respectively.</i>	12
2.2	<i>The half planes associated with an edge e.</i>	13
2.3	<i>An edge splits the plane into regions of different valid orientations for cast removal</i>	20
2.4	<i>The center r is located in the black region of the edge containing c_1. .</i>	22
2.5	<i>Location of a point r in a bounded region.</i>	23
2.6	<i>Result of the right envelope algorithm (shown in bold)</i>	24
2.7	<i>The near cone and the far cone of v.</i>	27
2.8	<i>The center r is located in the black region of the reflex vertex c_1. . . .</i>	29
2.9	<i>Approximation of half circle</i>	31
2.10	<i>Polygonal chain</i>	32
2.11	<i>Example with $s = 10$. (a) shows the polygon, (b) shows an enlargement of the polygonal chain c_2.</i>	33

3.1	<i>Two different views of P are shown along with the partitioning of P. The components P_1 and P_2 obstruct each other's rotational paths. . .</i>	36
3.2	<i>The prisms $S^+(f)$ and $S^-(f)$.</i>	38
3.3	<i>Left: The near and far wedge of e. Right: The near and far cone of v.</i>	58
3.4	<i>Example of polyhedron P that is not castable with respect to l although l does not intersect the black regions induced by P. Figure shows per- spective view and front view.</i>	68

List of Notation

- P polyhedron of arbitrary genus, 36
 P simple polygon, 13
 \cdot dot product of two vectors, 38
 ∂P boundary of P , 13, 36
 \times cross product of two vectors, 38
 \vec{l} direction of the line l in 3-dimensional space, 38
 \vec{n} inner normal of the face f , 40
 \vec{p} position vector of the point p , 40
 $int(P)$ interior of P , 13, 36
 n combinatorial complexity of the polyhedron P , 36
 n number of vertices of the simple polygon P , 13
 $CCW(f)$ set of edges of the face f removable in ccw orientation with respect to the line l , 49
 $CW(f)$ set of edges of the face f removable in cw orientation with respect to the line l , 49
 $FC(v)$ far cone of the vertex v , 28
 $NC(v)$ near cone of the vertex v , 28
 $S(f)$ open unbounded prism induced by the face f , 40
 $S^+(e)$ near wedge of the edge e , 60
 $S^+(f)$ black region of the face f , 40, 59
 $S^+(v)$ near cone of the vertex v , 60
 $S^-(e)$ far wedge of the edge e , 60
 $S^-(f)$ $S(f) \setminus S^+(f)$, 40
 $S^-(v)$ far cone of the vertex v , 60
 \vec{pr} vector from the point p to the point r , 16
 c near point with respect to a point r in the plane, 18
 $ccwarc(r, p, \alpha)$ circular arc with center point r and angle α starting at point p winding in ccw direction, 14
 $cl(S)$ closure of the open set S , 15
 $cwarc(r, p, \alpha)$ circular arc with center point r and angle α starting at point p winding in cw direction, 14
 $deg(f)$ degree of the face f , 63
 $deg(v)$ degree of the vertex v , 63
 e edge of P , 14, 43, 60
 f face of P , 37
 f far point with respect to a point r in the plane, 18
 $l(e)$ supporting line of the edge e , 15

$l^+(e)$ open half plane bounded by $l(e)$
located to the left of the edge e
when traversing the simple poly-
gon P in ccw orientation, 15

$l^-(e)$ open half plane bounded by $l(e)$
located to the right of the edge
 e when traversing the simple poly-
gon P in ccw orientation, 15

$n_e(a)$ line perpendicular to the edge e
passing through the vertex a of
 e , 15

$n_e^+(a)$ open half plane bounded by $n_e(a)$
containing the edge e , 15

$n_e^-(a)$ open half plane bounded by $n_e(a)$
not containing the edge e , 15

v vertex of P , 28, 60

cast boundary of P , 13, 36

ccw counterclockwise, 14

cw clockwise, 14

Chapter 1

Introduction

Ubi materia, ibi geometria.

(Wherever there is matter there is also geometry.)

JOHANNES KEPLER

1.1 Motivation

Casting is an old and popular manufacturing technique, where liquid is poured into an open volume bounded by cast parts and the cast is removed once the liquid has hardened. Removing the cast from the manufactured object does not result in breaking the cast and hence, identical objects can be produced. Casting has a long history; the first sand casts were used to produce weapons made of copper about 6000 years ago [19]. The oldest known casting that still exists today is a frog out of copper that was produced around 3200 B.C. [37] (see Figure 1.1). Throughout the course of history, humans developed skills to use various materials and methods for casting. An elephant out of bronze cast in two parts is shown in Figure 1.1.

The development of casting as a manufacturing technique had a significant influence on raising living standards in history, because casting made it possible to manufacture affordable stoves and cars for virtually everybody. According to Simpson [37], casting is still one of the most commonly used manufacturing processes today. In fact, there exist product components, such as car engine blocks, that can

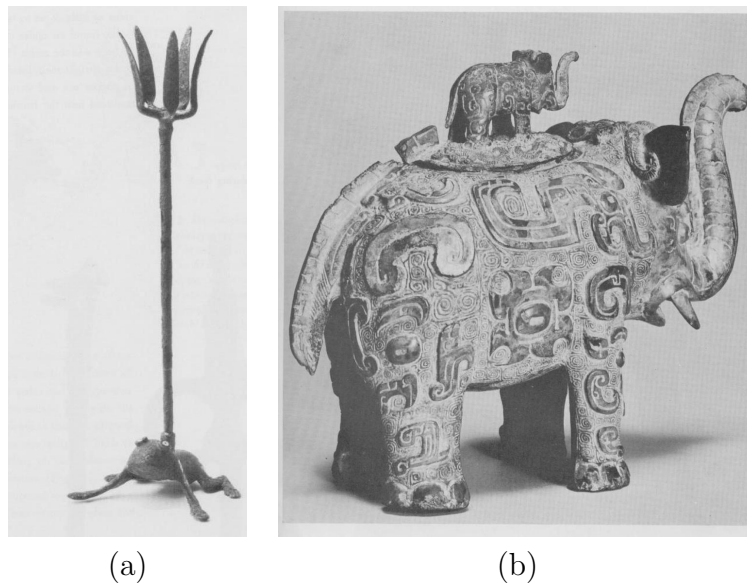


Figure 1.1: *Figure (a): Copper frog cast around 3200 B.C. Figure (b): Bronze elephant cast around 1122 to 255 B.C. Figures out of Simpson's book [37].*

only be manufactured through casting.

Casting is a complex manufacturing process and introduces various problems that are subject to research. These problems include, but are not limited to, the functional design of the casting process, where one considers the shape of the cast, the gate, and the feeding systems, thermodynamic effects and shrinkage during the solidification process, and fluid effects when the mold is filled. In this thesis, only the shape of the cast is considered. A given cast must be removable from the manufactured object without breaking the object or the cast. Only if this condition holds can the object be manufactured and the cast reused for mass production. However, it is only one preliminary step to determine whether the object is castable; physically casting the object involves the physical considerations stated above. An extensive overview of problems related to the casting process can be found in the book by Creese [19].

In the context of manufacturing, the term casting usually refers only to metal casting. Hence, producing plastic parts by pouring liquid into a cast is not included into casting in that context. The reason is that there are significant differences between the two manufacturing processes when shrinkage, fluid effects and thermodynamic

effects are considered. However, the geometric aspects of the two manufacturing processes are identical. Therefore, in computational geometry casting refers to any manufacturing process where liquid is poured into a cast and the cast is removed once the liquid has hardened. Hence, the results in this thesis can be applied to various areas including metal casting, producing plastic objects using a mold, or chocolate bunny production.

1.2 Problem Statement

In this thesis, the geometric aspects of *clamshell casting* are considered in two and three dimensions. The following two paragraphs explain what we mean by clamshell casting in both two and three dimensions.

In two dimensions, assume that we wish to manufacture an object modeled by a simple polygon P with n vertices. Let the boundary of P be the *cast* of P . An illustration can be found in Figure 1.2(a). Two problems are addressed. First, given a center of rotation r in the plane, determine whether there exists a partitioning of the cast into exactly two connected parts, such that one part can be rotated clockwise around r and the other part can be rotated counterclockwise around r without colliding with the interior of P . We present an algorithm to solve this problem with running time $O(n)$. Second, two algorithms are presented to find all the points in the plane that allow a cast partitioning as described above for convex and simple polygons, respectively. The running time of the algorithm for convex polygons is $O(n)$. For simple polygons with reflex vertices, the algorithm takes time $O(n^2)$. We provide an $\Omega(n^2)$ lower bound thereby proving the optimality of the algorithm.

In three dimensions, assume that we wish to manufacture an object modeled by a polyhedron P with combinatorial complexity n . Let the boundary of P be the cast of P . An illustration can be found in Figure 1.2(b). Two problems are addressed. First, given a line of rotation l in space, we determine whether there exists a partitioning of the cast into exactly two connected parts, such that one part can be rotated clockwise around l and the other part can be rotated counterclockwise around l without colliding with the interior of P or the cast. We present an algorithm to solve this problem with

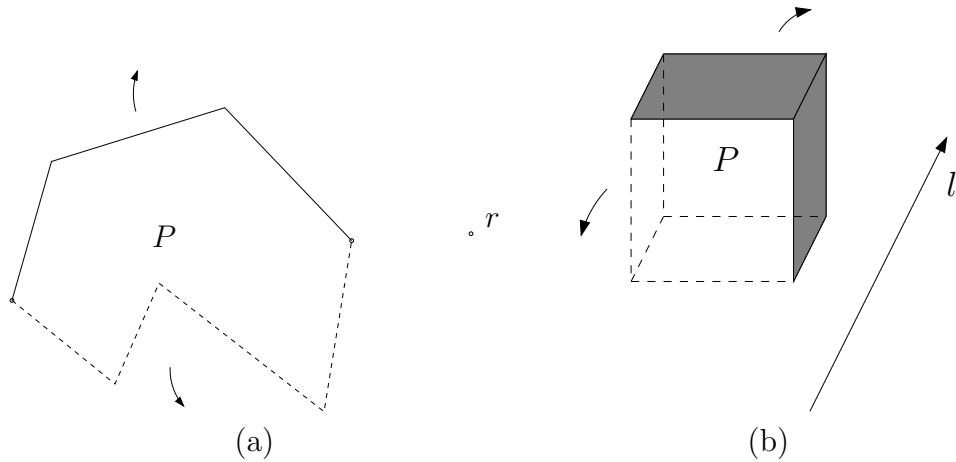


Figure 1.2: *Figure (a): The polygon P is partitioned into exactly two connected chains, such that the solid chain can be rotated clockwise around r and the dashed chain can be rotated counterclockwise around r without intersecting the interior of P . Figure (b): The polyhedron P is partitioned into exactly two connected parts, such that the shaded part can be rotated clockwise around l and the white part can be rotated counterclockwise around l without intersecting the interior of P .*

running time $O(n \log n)$. If we restrict the problem further and determine whether such a partitioning is possible when no reflex edge or face of P is perpendicular to l , the algorithm runs in $O(n)$ time.

Second, an algorithm is presented to find all the lines in space that allow a cast partitioning as described above. The algorithm's running time is $O(n^4 \log n)$. Again, we can restrict the problem further and find all the lines in space that allow a cast partitioning as described above, such that no reflex edge or face of P is perpendicular to l . In this case, the algorithm's running time is $O(n^4 \alpha(n))$, where $\alpha(n)$ is the inverse Ackermann function. Alternatively, P can be preprocessed in $O(n^{4+\epsilon})$ expected time into a data structure of size $O(n^{4+\epsilon})$, such that for any given line l , we can decide in $O(\log n)$ time if l is a valid casting line.

1.3 Related Work

The problem of determining whether a given object modeled by a polygon in two dimensions or a polyhedron in three dimensions, respectively, can be manufactured

using the casting process is a well-studied problem in computational geometry. In fact, the problem is discussed in Chapter 4 of the textbook by de Berg et al. [21]. The following overview of related problems is not extensive. For a detailed discussion of problems related to manufacturing processes considered in computational geometry, the reader is referred to Bose [5], Bose and Toussaint [9], and the Handbook of Discrete and Computational Geometry [24].

This thesis is concerned with rotational casting problems in two and three dimensions, where the object to be manufactured is modeled by a polygon or polyhedron, respectively. The cast consists of exactly two parts and is removed using a rotation around a point in the plane or a line in space, respectively. To my knowledge, this problem has not been considered in the past, but has been stated as an open problem in different publications [5, 6, 35]. However, the analogous translational casting problem, where the two parts of the cast are removed using a translation has been studied extensively in both two and three dimensions [1, 2, 6, 26, 35].

It is important to study the geometric setting of rotational casting problems, since molds designed for rotational casting are *permanent molds* [19]. That is, molds designed for rotational casting are not destroyed by the casting process and are reused for mass production. The rotational casts studied in this thesis consist of two cast parts that are attached by a hinge. This hinge is the main reason why the molds can be reused for mass production, since it is easy to align the two cast pieces when closing the cast. Note that molds designed for translational casting do not have this feature. Therefore, molds designed for translational casting are usually *consumable molds* that are destroyed by the casting process [19].

There exists a close relationship between rotational casting and translational casting. Let P denote a polygon in 2-dimensional space and let r denote the center of rotation used to rotationally cast P . Assume that P and r are given in a polar coordinate system with origin r . In polar coordinates, every point is described by an angle ϕ and the distance d from the origin. Transform the coordinate system into a cartesian coordinate system, such that the x -axis describes the angle ϕ and the y -axis describes the distance d . Considering this transformed system between $x = 0$ and $x = 2\pi$ shows the transformed polar coordinate system. Every point of the cast

of P moves along a straight line when the cast is removed. This means that rotational casting becomes translational casting. It remains to analyze the shape of the transformed polygon \tilde{P} . Without loss of generality (since everything can be rotated), assume that P does not contain vertical edges. Furthermore, assume that r is not contained in the interior of P , since otherwise, P could not be cast using r as center of rotation. The transformation of an edge e of P is in essence a curve describing the distance of points on the edge to the origin. The distance from an edge to a point is non-linear and cannot be described using an algebraic curve, but trigonometric functions are necessary [4]. Since r is not contained in the interior of P , the transformed polygon \tilde{P} is topologically equivalent to P and its boundary consists of piecewise non-algebraic curves. Considering rotational casting of a polygon P is equivalent to considering translational casting of the transformed polygon \tilde{P} .

The above relationship between rotational casting and translational casting extends to three dimensions. Let P denote a polyhedron and let l denote the axis of rotation. Assume that P and l are given in a cylindrical coordinate system with l as z -axis. Transform the coordinate system into a cartesian coordinate system, such that the x -axis describes the angle ϕ , the y -axis describes the distance d , and the z -axis stays fixed. This yields the transformed polyhedron \tilde{P} bounded by non-algebraic surface patches. Furthermore, rotations around l correspond to translations in the transformed coordinate system. Hence, the following result can be extended: Considering rotational casting of a polyhedron P is equivalent to considering translational casting of the transformed polyhedron \tilde{P} .

To my knowledge, this close relationship between rotational casting and translational casting has not previously been mentioned or used to obtain algorithms to rotationally cast polygons or polyhedra. None of the algorithms surveyed in the following can easily be extended to handle translational castability of 2-dimensional objects bounded by piecewise non-algebraic curves or 3-dimensional objects bounded by non-algebraic surface patches.

In two dimensions, Rosenbloom and Rappaport [35] consider an object modeled by a simple polygon with n vertices and they define the cast of this object as the boundary of the polygon. Their algorithms solve two related problems. The first

algorithm determines in time $O(n)$ whether the cast can be partitioned into exactly two pieces, such that each piece can be removed from the manufactured object by translation without breaking the object or the cast. To solve this problem, an important link between castability and monotonicity of a simple polygon is established. The algorithm that solves this problem uses the algorithm developed by Preparata and Supowit [34] that determines in $O(n)$ time whether a polygon with n vertices is monotone. The second algorithm determines in time $O(n \log n)$ whether the cast can be partitioned into two pieces by cutting the cast along a line, such that both pieces can be removed from the manufactured object by translation without breaking the object or the cast. If this is possible, the object can be manufactured in the following way: the cast is cut into two pieces and the two pieces are put on their sides and filled with liquid. Once the liquid has hardened, the cast is removed by translation and the two manufactured pieces are glued together. This method of manufacturing an object is called *sand-casting*.

Different approaches exist to examine the three-dimensional version of the casting problem, where the object to be manufactured is modeled by a polyhedron in space with combinatorial complexity n and the polyhedron's boundary is used as cast. As mentioned above, these approaches are closely related to rotational clamshell casting. Ahn et al. [2] determine whether the cast can be partitioned into exactly two pieces, such that both pieces can be removed from the manufactured object by translations in opposite directions without breaking the object or the cast. They consider two problems: the first problem is to decide whether a given object is castable with respect to a given cast removal direction and the second problem is to find a valid removal direction for a given object in case that one exists. Ahn et al. give sufficient and necessary conditions for translational castability of objects bounded by algebraic surface patches that meet along algebraic curves. However, algorithms to determine castability only work for polyhedra. For a polyhedron of combinatorial complexity n , they present algorithms to solve the first problem in $O(n \log n)$ time and to solve the second problem in $O(n^4)$ expected time. They also show that their algorithm cannot be easily extended to allow arbitrary cast removal directions, i.e. non-opposite directions. Note that the algorithm provided by Ahn et al. solves a three-dimensional

version of the first problem considered by Rosenbloom and Rappaport. Before Ahn et al. found an analytic solution to this problem, Hui and Tan [26] developed a heuristic method to solve the problem. They choose candidate removal directions heuristically and test for a sample of points located on the boundary of the polyhedron whether the points can be removed in that direction. Ahn et al. [1] solve the two above-mentioned problems using randomized algorithms in a setting, where the direction for cast removal is uncertain. This setting is important in practice, since the control of the casting machinery is imperfect.

Note that the 3-dimensional problems considered in this thesis are closely related to the problems considered by Ahn et al. in [2]. The 3-dimensional decision problem for translational casting considered by Ahn et al. corresponds to the 3-dimensional decision problem addressed in this thesis. The problem of determining all the valid removal directions considered by Ahn et al. corresponds to the problem of determining all valid casting lines in space addressed in this thesis. Since Ahn et al. propose algorithms that handle opposite cast removal only, the space of solutions is the space of all possible directions, i.e. the unit sphere of directions, and therefore 2-dimensional. The problem considered in this thesis considers full lines. Hence the space of all solutions is the space of all possible lines in 3D and is therefore 4-dimensional [33]. Note that despite these two additional degrees of freedom, the algorithms presented in this thesis are only slightly less efficient than the algorithms proposed by Ahn et al.

Bose et al. [6] consider an object modeled by a simple polyhedron with n vertices and use the polyhedron's boundary as cast. They determine whether the cast can be partitioned into two pieces by a plane, such that both pieces can be removed from the manufactured object by translation without breaking the object or the cast. If this is the case, the object can be manufactured by sand casting. A simple algorithm to decide whether a simple polyhedron with n vertices can be manufactured by sand casting running in time $O(n^2 \log n)$ is provided. The running time of the algorithm can be improved to $O(n^{\frac{3}{2}+\epsilon})$, for any fixed $\epsilon > 0$, by using complicated data structures. Furthermore, the running time of the algorithm becomes $O(n^2)$ in case that the cast removal directions are opposite directions. Note that the algorithm provided by

Bose et al. solves a three-dimensional version of the second problem considered by Rosenbloom and Rappaport.

In computational geometry, other problems related to the casting process are subject to research. For instance, when a polyhedral shape is manufactured through casting, the cast has at least one hole called a *filling hole* through which molten liquid is poured into the cast. Furthermore, the cast may contain additional holes called *venting holes* that allow the air initially contained in the cast to escape as the cast is filled with liquid. Venting holes are crucial to eliminate surface defects and the number of venting holes depends on the orientation of the cast. Bose et al. [12] describe an algorithm running in time $O(n^2)$ that computes the orientation of the cast, modeled by a polyhedron with n vertices, that minimizes the number of venting holes required to eliminate surface defects. More efficient algorithms to solve the problem are presented for special classes of polyhedra. The two-dimensional version of this problem is addressed by Bose and Toussaint [10] and Fekete and Mitchell [22].

1.4 Summary of Results and Thesis Organization

Chapter 2 discusses the two-dimensional clamshell casting problem. Two problems are addressed: finding a partitioning of the cast based on a given point of rotation and finding all of the points in the plane that allow a valid partitioning of the cast. A preliminary version of Chapter 2 was published by Bose et al. [8]. Table 1.1 summarizes the results presented in Chapter 2.

Chapter 3 discusses the three-dimensional clamshell casting problem. Two problems are addressed: finding a partitioning of the cast based on a given line of rotation and finding all of the lines in space that allow a valid partitioning of the cast. A preliminary version of Chapter 3 was published by Bose et al. [7]. Table 1.2 summarizes the results presented in Chapter 3.

Finally, Chapter 4 concludes and gives ideas for future work.

	Convex Polygon	Simple Polygon
Decision Problem	$O(n)$ time $O(n)$ space	$O(n)$ time $O(n)$ space
Reporting all valid centers of rotation	$O(n)$ time $O(n)$ space	$O(n^2)$ time $O(n^2)$ space
Preprocessing polygon for queries	$O(n)$ preprocessing time $O(n)$ space $O(\log n)$ query time	$O(n^2)$ preprocessing time $O(n^2)$ space $O(\log n)$ query time

Table 1.1: *Summary of the results for two-dimensional clamshell casting.*

	Robust Castability	General Castability
Decision Problem	$O(n)$ time $O(n)$ space	$O(n \log n)$ time $O(n)$ space
Reporting all valid casting lines	$O(n^4 \alpha(n))$ time $O(n^4)$ space	$O(n^4 \log n)$ time $O(n^4)$ space
Preprocessing polyhedron for queries	$O(n^{4+\epsilon})$ preprocessing time $O(n^{4+\epsilon})$ space $O(\log n)$ query time	$O(n^{4+\epsilon})$ preprocessing time $O(n^{4+\epsilon})$ space $O(\log n)$ query time

Table 1.2: *Summary of the results for three-dimensional clamshell casting.*

Chapter 2

Rotational Casting in Two Dimensions

The world is continuous, but the mind is discrete.

DAVID MUMFORD

This chapter discusses the two-dimensional clamshell casting problem. Section 2.1 introduces the notation and preliminaries used throughout the chapter, Section 2.2 discusses the problem of finding a partitioning of the cast based on a given point of rotation, and Section 2.3 discusses the problem of finding all of the points in the plane that allow a valid partitioning of the cast.

2.1 Preliminaries

Let P be a simple polygon in the plane with n vertices and let $int(P)$ and ∂P denote the interior and boundary of P , respectively, so that $P = int(P) \cup \partial P$. The boundary is also called the *cast* of P . The edges of P are oriented counterclockwise so that $int(P)$ is located to their left. Parallel adjacent edges are not allowed, since this can be easily avoided by merging the two adjacent parallel edges. The aim is to rotationally remove the cast of P in two pieces. We specify below precisely what this means.

Definition 2.1.1. Let r and p be points in the plane. Denote the circular arc with center r and angle α starting at p winding in clockwise (cw) or counterclockwise (ccw) direction by $cwarc(r, p, \alpha)$ or $ccwarc(r, p, \alpha)$ respectively. An edge e of P is removable in cw orientation with respect to r if

$$\exists \alpha > 0 \text{ such that } \forall p \text{ on } e : cwarc(r, p, \alpha) \cap \text{int}(P) = \emptyset$$

and removable in ccw orientation with respect to r if

$$\exists \alpha > 0 \text{ such that } \forall p \text{ on } e : ccwarc(r, p, \alpha) \cap \text{int}(P) = \emptyset.$$

Then, we call the cw or ccw orientation a *valid orientation for cast removal for e with respect to r* respectively, and we call r a *valid center of rotation for e* . Figure 2.1 illustrates the definition of removability for edges.

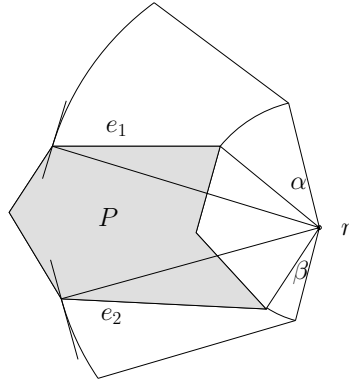


Figure 2.1: The edges e_1 and e_2 are removable in cw orientation with angle α and ccw orientation with angle β with respect to r respectively.

Definition 2.1.2. Let r be a point in the plane. A polygon P is rotationally castable with respect to r , if ∂P can be partitioned into exactly two connected chains, such that all edges of one chain are removable in cw orientation with respect to r and all edges of the other chain are removable in ccw orientation with respect to r .

This implies that there exists an angle α , such that both chains can be rotated by angle α in cw or ccw orientation with respect to r , respectively, without colliding with each other. Note that the partitioning of the chain is not necessarily at vertices of P .

Henceforth, unless specifically stated otherwise, by *castable*, we will mean rotationally castable. We now outline a key property that characterizes all locations from which an edge is removable.

For an edge $e \in \partial P$ with incident vertices a and b , let $n_e(a)$ denote the line perpendicular to e passing through a . The line $n_e(a)$ divides the plane into two half planes and the open half plane containing b is denoted by $n_e^+(a)$ and the open half plane that does not contain b is denoted by $n_e^-(a)$. The supporting line $l(e)$ of e divides the plane into two half planes. Denote the open half plane located to the left of e when traversing P in ccw orientation by $l^+(e)$ and the open half plane located to the right of e when traversing P in ccw orientation by $l^-(e)$, see Figure 2.2. The closure of an open set S is denoted by $cl(S)$.

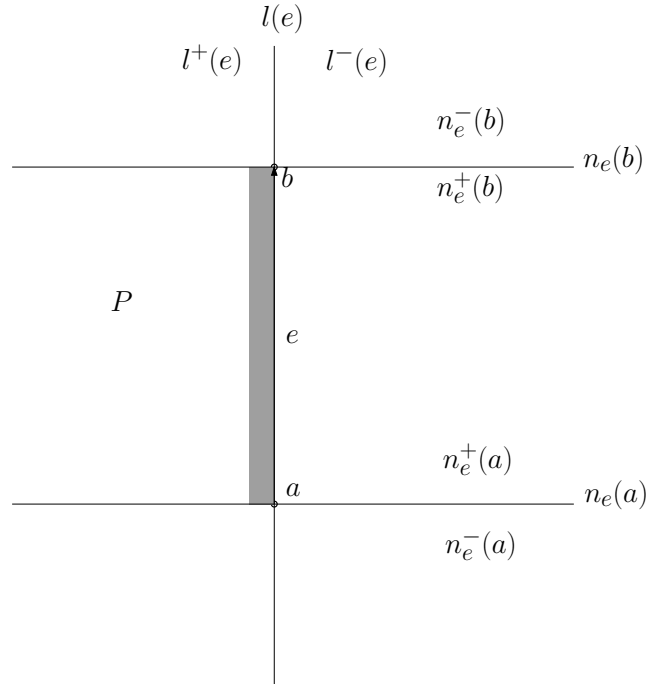


Figure 2.2: The half planes associated with an edge e .

Lemma 2.1.3. *Let e be an edge of P and denote the two vertices incident to e in ccw order by a and b . For the orientation for cast removal of e , the following four cases are possible:*

1. The edge e is removable from the cast using a cw rotation around r , if and only if $r \in cl(n_e^-(a))$.
2. The edge e is removable from the cast using a ccw rotation around r , if and only if $r \in cl(n_e^-(b))$.
3. The edge e needs to be partitioned into two parts at the orthogonal projection of r on e in order to be removed, if and only if $r \in n_e^+(a) \cap n_e^+(b) \cap cl(l^-(e))$. One part of e is removable using a ccw rotation and the other one using a cw rotation around r . Let r^* be the orthogonal projection of r on e . Denote the edge with incident vertices a and r^* by e_1 and the edge with incident vertices r^* and b by e_2 respectively. The edge e_1 is removable using a ccw rotation around r and e_2 is removable using a cw rotation around r .
4. The edge e is not removable from the cast, if and only if $r \in n_e^+(a) \cap n_e^+(b) \cap l^+(e)$.

Proof. Consider that every point p of e moves on $cwarc(r, p, \alpha)$ or $ccwarc(r, p, \alpha)$ when rotated by an angle α around r . Denote the vector from p to r by \vec{pr} and the vector \vec{pr} rotated in ccw orientation by 90° by \vec{pr}^\perp . The tangent of $cwarc(r, p, \alpha)$ or $ccwarc(r, p, \alpha)$ is \vec{pr}^\perp or $-\vec{pr}^\perp$, respectively, for any point p . Denote by g the line passing through p in direction \vec{pr}^\perp and denote by $g^+(p)$ the closed half plane bounded by g containing r . The two arcs $cwarc(r, p, \alpha)$ and $ccwarc(r, p, \alpha)$ are contained in $g^+(p)$.

Let p be an arbitrary point in the interior of e . There exists an open disk d with positive radius centered at p with the property that exactly half of d is contained in $int(P)$ and exactly half of d is contained in the exterior of P . Denote the ray starting at p propagating in direction \vec{pr}^\perp by q^+ and denote the ray starting at p propagating in direction $-\vec{pr}^\perp$ by q^- .

Let $r \in cl(n_e^-(a))$ and let p be an arbitrary point in the interior of e . The intersection $d \cap q^+$ is located completely outside of $int(P)$. Hence, p can move by a small amount along \vec{pr}^\perp without penetrating $int(P)$. Since $cwarc(r, p, \alpha) \subseteq g^+(p)$ and since \vec{pr}^\perp is the tangent of $cwarc(r, p, \alpha)$ in p , small movements of p along $cwarc(r, p, \alpha)$ are possible without penetrating $int(P)$. Hence, $\exists \alpha > 0$ such that $\forall p$ on e :

$cwarc(r, p, \alpha) \cap int(P) = \emptyset$. The intersection $d \cap q^-$ is completely contained in $int(P) \cup \{p\}$ and hence, p cannot move infinitesimally along $-\vec{p}r^\perp$ without penetrating $int(P)$. Since infinitesimal movements along $-\vec{p}r^\perp$ correspond to infinitesimal movements along $ccwarc(r, p, \alpha)$, there is no $\alpha > 0$ such that $\forall p \text{ on } e : ccwarc(r, p, \alpha) \cap int(P) = \emptyset$. Hence, e is only removable using a cw rotation around r if $r \in cl(n_e^-(a))$.

Let $r \in cl(n_e^-(b))$ and let p be an arbitrary point in the interior of e . The intersection $d \cap q^-$ is located completely outside of $int(P)$. Hence, p can move by a small amount along $-\vec{p}r^\perp$ without penetrating $int(P)$. Since $ccwarc(r, p, \alpha) \subseteq g^+(p)$ and since $-\vec{p}r^\perp$ is the tangent of $ccwarc(r, p, \alpha)$ in p , small movements of p along $ccwarc(r, p, \alpha)$ are possible without penetrating $int(P)$. Hence, $\exists \alpha > 0$ such that $\forall p \text{ on } e : ccwarc(r, p, \alpha) \cap int(P) = \emptyset$. The intersection $d \cap q^+$ is completely contained in $int(P) \cup \{p\}$ and hence, p cannot move infinitesimally along $\vec{p}r^\perp$ without penetrating $int(P)$. Since infinitesimal movements along $\vec{p}r^\perp$ correspond to infinitesimal movements along $cwarc(r, p, \alpha)$, there is no $\alpha > 0$ such that $\forall p \text{ on } e : cwarc(r, p, \alpha) \cap int(P) = \emptyset$. Hence, e is only removable using a ccw rotation around r if $r \in cl(n_e^-(b))$.

If $r \in n_e^+(a) \cap n_e^+(b) \cap cl(l^-(e))$, e is divided into two edges at the orthogonal projection r^* of r on e . Denote the edge with incident vertices a and r^* by e_1 and the edge with incident vertices r^* and b by e_2 respectively. As $r \in cl(n_{e_1}^-(r^*))$ and $r \in cl(n_{e_2}^-(r^*))$, e_1 is only removable using a ccw rotation around r and e_2 is only removable using a cw rotation around r .

If $r \in n_e^+(a) \cap n_e^+(b) \cap l^+(e)$, the orthogonal projection r^* of r on e cannot be rotationally removed from the cast. This means, there is no $\alpha > 0$ such that $cwarc(r, r^*, \alpha) \cap int(P) = \emptyset$ or $ccwarc(r, r^*, \alpha) \cap int(P) = \emptyset$ respectively. Therefore, e is not removable with respect to r .

This determines the removability of e depending on the location of r in the plane. Hence, the four statements of Lemma 2.1.3 follow directly. \square

2.2 Decision Problem

In this section, we address the question of whether a polygon is castable with respect to a given point of rotation and present an algorithm that solves the problem in linear time.

Assume that a polygon P and a center of rotation r are given. The aim is to determine whether P is castable with respect to r . If P is castable with respect to r then the two points on ∂P , where the cast is cut, need to be found.

Definition 2.2.1. *A near point c with respect to r is defined as $c \in \partial P$ with the property that a sufficiently small neighborhood of c on ∂P is completely outside of the open disk centered at r and passing through c . This means there exists a disk b centered at c with a positive radius, such that all points $q \in (\partial P \cap b) \setminus \{c\}$ are outside of the closed disk centered at r and passing through c .*

Hence, if c is not a vertex, c is the orthogonal projection of r on an edge of P . Therefore, c locally minimizes the distance between the cast and the center of rotation r .

Definition 2.2.2. *A far point f with respect to r is defined as $f \in \partial P$ with the property that a sufficiently small neighborhood of f on ∂P is completely contained in the closed disk centered at r and passing through f . This means there exists a disk b centered at f with a positive radius, such that all points $q \in \partial P \cap b$ are completely contained in the closed disk centered at r and passing through f .*

The point f is always a vertex of P that locally maximizes the distance between the cast and the center of rotation r .

Definition 2.2.3. *Let $p \in \partial P$ be a vertex of P or a point in the interior of an edge of P . If p is located in the interior of an edge e , split the edge into two edges at p . The valid orientation for cast removal with respect to r is said to change at p if one of the edges incident to p is removable in cw orientation and the other edge incident to p is removable in ccw orientation with respect to r .*

Lemma 2.2.4. *The valid orientation for cast removal with respect to r changes at a point $p \in \partial P$ if and only if p is either a near point or a far point with respect to r .*

Proof. The proof consists of two parts. First, we show that the valid orientation for cast removal with respect to r changes at $p \in \partial P$ if p is a near point or a far point with respect to r . At a far point f , an arbitrarily small neighborhood of f is completely contained in the closed disk induced by the circle b centered at r passing through f . Hence, there is a smaller circle concentric to b that passes through two points on ∂P located in an arbitrarily small neighborhood of f . As this circle intersects the polygon twice, one intersection point penetrates $\text{int}(P)$ when rotated by an arbitrarily small angle in cw orientation with respect to r and the other intersection point penetrates $\text{int}(P)$ when rotated by an arbitrarily small angle in ccw orientation with respect to r . Hence, it is not possible to remove the cast in the same orientation. Hence, the valid orientation for cast removal changes at f . The proof is similar for near points where b is enlarged by a small amount. Again, the two intersection points of the enlarged circle with the polygon can only be removed in different orientations with respect to r .

Second, the valid orientation for cast removal with respect to r changes at no other point but a near point or a far point. Assume that the valid orientation for cast removal with respect to r changes at $p \in \partial P$ with p neither a far point nor a near point. Hence, the circle b centered at r passing through p properly intersects P at p , since p neither locally maximizes nor locally minimizes the distance between ∂P and r . If p is not a vertex of P , but situated in the interior of an edge e of ∂P , e is split into two edges at p . Otherwise, p is a vertex of P and there exist exactly two edges adjacent to p . Therefore, the point p has two adjacent edges. As P is a simple polygon, locally it is located completely to the left of the boundary defined by the two edges adjacent to p . Hence, the valid orientation for cast removal with respect to r does not change at p , which contradicts the initial assumption. Therefore, p must be either a near point or a far point for the valid direction of cast removal with respect to r to change. \square

Theorem 2.2.5. *Given a center of rotation r , a polygon P is castable with respect*

to r if and only if there exists exactly one near point c with respect to r and exactly one far point f with respect to r on ∂P .

Proof. The proof consists of two parts. First, we show that P is castable with respect to r if there exists exactly one near point c and exactly one far point f with respect to r . If there exists exactly one near point c and exactly one far point f with respect to r , the point c minimizes the distance between ∂P and r and f maximizes the distance between ∂P and r . Hence, P is completely contained in the closed annulus defined by the two concentric circles centered at r and passing through c and f respectively. The valid orientation for cast removal with respect to r can only change at c and f (Lemma 2.2.4). Therefore, one part of the polygon can be removed using a cw rotation and the other part can be removed using a ccw rotation if P is cut at c and f .

Second, if P is castable with respect to r then there exists exactly one far point and exactly one near point with respect to r . A castable polygon with respect to r consists of two cast parts, i.e. there are exactly two points on ∂P where the orientation of cast removal with respect to r changes. By Lemma 2.2.4, this implies that there are exactly two near or far points on ∂P . The extreme value theorem implies that there is always at least one local minimum and one local maximum with respect to the distance from r to ∂P [28, Chapter 3]. Therefore, there must exist at least one near point and one far point $\in \partial P$ with respect to r . Hence, there is exactly one near point and one far point with respect to r on a castable polygon with respect to r . \square

Theorem 2.2.5 allows us to determine whether a polygon is castable given a center point r by testing how many points $p \in \partial P$ are local extrema with respect to the distance between p and r . The polygon is castable if and only if there is exactly one maximum and one minimum. To do this test efficiently, we establish the following observation:

Observation 2.2.6. *For a simple polygon P and a point r in the plane, the number of points $c \in \partial P$ that locally minimize the distance between ∂P and r equals the number of points $f \in \partial P$ that locally maximize the distance between ∂P and r .*

Observation 2.2.6 holds because P is a simple closed polygon. Hence, the function describing the distance from r to ∂P is continuous and there is always a local minimum between two local maxima and vice versa for continuous functions [28, Chapter 3].

Hence, it is sufficient to consider local maxima to decide whether a polygon is castable given a center point r . As each far point must be a vertex of P , one can test for multiple local maxima by traversing the polygon's vertices p and computing the distances between p and r . Furthermore, it is required to compute the minimal distance from r to each edge of P . The reason is that two consecutive vertices of P can both locally maximize the distance between r and ∂P . This takes linear time.

Theorem 2.2.7. *Given a polygon P with n vertices and a center of rotation r in the plane, we can test in $O(n)$ time whether P is castable with respect to r .*

2.3 Determining all valid regions for cast removal

In this section, the aim is to find all points r in the plane, such that a given polygon is castable with respect to r .

Definition 2.3.1. *The set of all points r in the plane with the property that P is castable with respect to r is the valid region for cast removal of P . The complement of the valid region is the invalid region for cast removal of P .*

The aim is to determine the valid region in the plane for a given polygon P by partitioning the plane into valid and invalid region for cast removal. Once a query point r is given, it is possible to determine whether r is a valid center of rotation for P by determining whether r is contained in a valid or an invalid region for cast removal. We will see that convex polygons have a valid region that differs significantly from the valid region of non-convex simple polygons.

2.3.1 Convex polygons

In this section, we consider convex polygons and show that it is possible to find the valid region for cast removal in linear time. The plane is partitioned into valid and

invalid regions for cast removal by constructing the envelope of an arrangement of half lines.

Lemma 2.1.3 implies that every edge e with incident vertices a and b given in ccw order on ∂P splits the plane into regions of different valid orientations for cast removal, see Figure 2.3.

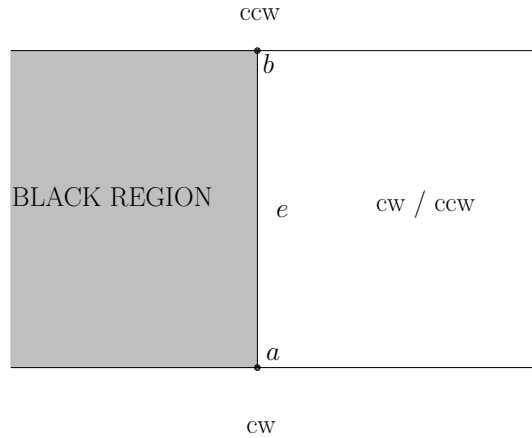


Figure 2.3: An edge splits the plane into regions of different valid orientations for cast removal

Definition 2.3.2. Let e be an edge of P and denote the two vertices incident to e in ccw order by a and b . The open strip $n_e^+(a) \cap n_e^+(b) \cap l^+(e)$ is called the black region of e .

Note that the black region does not contain any valid centers of rotation r for which e is castable (see Lemma 2.1.3, case 4).

Lemma 2.3.3. For a convex polygon P , $int(P)$ is contained in the union of the black regions of the edges of P .

Proof. Every point $q \in int(P)$ has at least one near point $c \in \partial P$ with respect to q . As P is convex and as $q \in int(P)$, c is the orthogonal projection of q on an edge e and not a vertex of P . Hence, q is contained in the black region of e . \square

Lemma 2.3.4. A convex polygon P is castable with respect to a center of rotation r if and only if r is not contained in the union of all black regions of edges of P .

Proof. This proof consists of two parts. First, a convex polygon is not castable with respect to r if r is contained in the union of all black regions of edges of P . If r is contained in the union of all black regions, it is contained in the black region of at least one edge e . The edge e is therefore not removable with respect to r (Lemma 2.1.3).

The second part is that P is castable with respect to r if r is not contained in the union of all black regions of edges of P . Assume, r is outside of the union of the black regions, and P is not castable. Theorem 2.2.5 and the Extreme Value Theorem [28, Chapter 3] imply that there are at least two far points with respect to r . Denote the two far points by f_1 and f_2 . Two cases can occur: either $r \in \text{int}(P)$ or $r \notin \text{int}(P)$. If $r \in \text{int}(P)$, Lemma 2.3.3 ensures that P is contained in the black region of at least one edge. Hence, $r \notin \text{int}(P)$ must hold. The following description is illustrated in Figure 2.4. Since $r \notin \text{int}(P)$, it is possible to compute two tangents from r to ∂P . Denote the two vertices where the tangents touch ∂P by t_1 and t_2 , respectively. If a tangent touches ∂P in more than one vertex, choose the vertex closest to r as t_1 or t_2 , respectively. The two tangents decompose ∂P into two chains, the lower chain contained in the triangle T with vertices t_1, t_2 , and r and the upper chain not contained in T . Since P is convex, no far point of P with respect to r can be on the lower chain. Hence, both f_1 and f_2 are on the upper chain. There are two near points on ∂P with respect to r , one on each chain connecting f_1 and f_2 . Since both f_1 and f_2 are on the upper chain, there must be a near point c_1 with respect to r on the upper chain between f_1 and f_2 (see Observation 2.2.6). Since P is convex and c_1 is on the upper chain, c_1 cannot be a vertex of P . Hence, c_1 is the orthogonal projection of r onto an edge e of P . Since e is on the upper chain and r projects orthogonally onto e , r is located to the left of e . Therefore $r \in n_e^+(a) \cap n_e^+(b) \cap l^+(e)$, where a and b denote the vertices incident to e . This means, r is contained in the black region of e . But this contradicts the initial assumption that r is not contained in the union of black regions of edges of P . Hence, P is only castable with respect to r if r is outside of the union of black regions of edges of P . \square

Lemma 2.3.5. *The valid region for cast removal of a convex polygon P consists only of unbounded regions in the plane.*

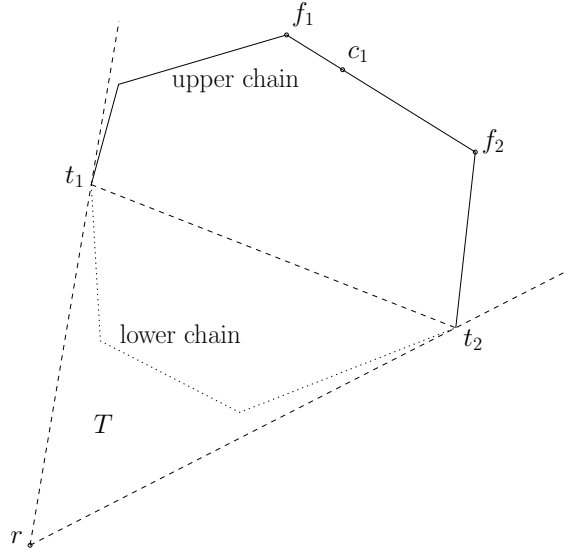


Figure 2.4: The center r is located in the black region of the edge containing c_1 .

Proof. Note that Lemma 2.3.4 implies that the complement of the union of the black regions of edges of a convex polygon P is the valid region for cast removal of P . Assume there exists a point r in a bounded region such that P is castable with respect to r . Then, r is contained in a region bounded by the black regions of at least two edges e_1 and e_2 of P and the convex polygonal chain h connecting e_1 and e_2 that has r to its left, see Figure 2.5. Let p_1 and p_2 be the vertices $e_1 \cap h$ and $e_2 \cap h$. The vertices p_1 and p_2 minimize the distance from r to e_1 and e_2 respectively. As the function describing the distance from r to ∂P is continuous and as P is simply connected, there exists at least one near point c with respect to r on h . As r is located to the left of h and as h is convex, c is located in the interior of an edge e with incident vertices a and b . Hence, $r \in n_e^+(a) \cap n_e^+(b) \cap l^+(e)$, i.e. r is contained in the black region of e . This contradicts the initial assumption and proves that the valid region of P consists only of unbounded regions in the plane. \square

Based on Lemma 2.3.4 and Lemma 2.3.5, we compute the boundary of the union of all black regions of edges of P . For this, the notion of an envelope of n lines is defined.

Definition 2.3.6. A set of n lines in the plane induces a subdivision S of the plane.

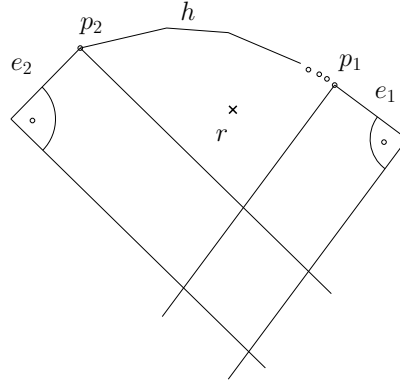


Figure 2.5: Location of a point r in a bounded region.

The envelope of the n lines is the polygon formed by the bounded edges of all the unbounded regions of S [27].

Similarly, a convex polygon P and the half lines bounding the black regions of its edges induce a subdivision S of the plane. Parallel half lines with the same orientation intersect at infinity and are therefore considered to be bounded edges. The polygon formed by the bounded edges of all the unbounded regions of S is called the envelope of the arrangement induced by P .

Lemma 2.3.5 implies that all valid regions of P are contained in the complement of the envelope of the arrangement induced by P .

Theorem 2.3.7. *Given a convex polygon P with n vertices, a description of the valid region for cast removal of P has $O(n)$ size and can be computed in $O(n)$ time.*

Proof. Using the algorithm of Keil [27], it is possible to compute the envelope of an arrangement of n lines in $O(n)$ time given that the lines are sorted according to their slope. This algorithm can be modified to find the union of all black regions of edges of P by defining an arrangement consisting of the half lines that bound black regions of edges. In this arrangement, the left and the right envelopes are computed, and their union corresponds to the union of all black regions of P . The modified algorithm first splits the polygon at the two points with minimum and maximum y -coordinate. The right envelope is computed by starting at the lowest point of the polygon and traversing it in clockwise order up to the highest point. For each edge e we traverse,

denote the half line in the direction of the inner normal of e passing through the first vertex of e encountered during the traversal by l_i and the half line in the direction of the inner normal of e passing through the second vertex of e encountered during the traversal by l_i^* , $1 \leq i \leq s, s < n$. See Figure 2.6. Denote by B_i the convex polygonal chain bounding the region below the half lines l_1 to l_i , $1 \leq i \leq s$, and by A_i the convex polygonal chain bounding the region above the lines l_{i+1}^* to l_s^* , $0 \leq i \leq s - 1$. Concatenate A_0 , for $1 \leq i \leq s - 1$ the boundary of $A_i \cap B_i$, B_s , and in case that A_0 and B_s are disjoint the part of P used to compute the right envelope. For a visualization of the result of this right envelope, refer to Figure 2.6.

To compute the left envelope, traverse the polygon in ccw direction starting at the lowest point and ending at the highest point. Define l_i and l_i^* , $1 \leq i \leq s, s < n$ as above for every edge of P . Computing A_i and B_i in the same way as before and concatenating A_0 , for $1 \leq i \leq s - 1$ the boundary of $A_i \cap B_i$, B_s , and in case that A_0 and B_s are disjoint the part of P used to compute the left envelope yields the left envelope. Note that the only difference between this algorithm and Keil's algorithm is the use of two different sets of lines l_i and l_i^* to compute B_i and A_i , respectively. Hence, only minor changes in Keil's algorithm are required to perform these computations. As there are $2n$ half lines already sorted by slope, this algorithm takes $O(n)$ time.

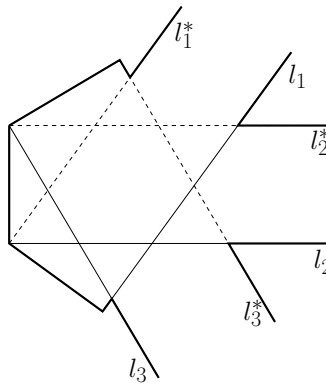


Figure 2.6: *Result of the right envelope algorithm (shown in bold)*

Two planar regions are created, and if we imagine that parallel lines intersect at infinity, the two regions are simply connected planar polygons. The algorithm

by Finke and Hinrichs [23], that computes the overlay of simply connected planar subdivisions in time linear in the size of the output, is used to compute the union of those two regions. The algorithm assumes that the two subdivisions are given in quad view data structure and changes that structure in a way that the result represents the overlay of the two regions.

The size of the two envelopes E_1 and E_2 is linear in the number n of vertices of the polygon P , because it can be computed using Keil's algorithm in $O(n)$ time. As both envelopes ordered in clockwise order are given, one can construct a quad view data structure in linear time. The time required for Finke and Hinrichs's algorithm is $O(n+k)$, where n is the combined size of the two polygons to be overlaid and k is the number of intersection points of E_1 and E_2 . Lemma 2.3.5 guarantees that there are no unbounded valid regions in the overlay of E_1 and E_2 . Hence, when an edge of E_1 intersects an edge of E_2 , only one of the edges can have further intersection points with E_1 or E_2 respectively. Therefore, the number of intersection points of E_1 and E_2 is $O(n)$ resulting in an $O(n)$ time algorithm. In the resulting subdivision, any region labeled as unbounded is a valid region of P .

The combination of the two algorithms allows to find the valid region for centers of rotations for cast removal in $O(n)$ time where n is the number of vertices of P . \square

Corollary 2.3.8. *A convex polygon P with n vertices can be preprocessed in $O(n)$ time, such that for any given point r , we can decide in $O(\log n)$ time if P is castable with respect to r .*

Proof. Theorem 2.3.7 allows to find the valid region for cast removal of P in $O(n)$ time. Hence, in $O(n)$ time, the plane is preprocessed, such that every face of the planar subdivision induced by black regions of P is labeled as a valid or invalid region.

For any query point r , after $O(n)$ preprocessing time, it is possible to determine the face of the arrangement containing r in time $O(\log n)$ [29]. Once the face is known, we can determine in constant time whether that face is contained in the union of the black regions of P , i.e. whether r is a valid center of rotation. \square

2.3.2 Simple polygons

In this section, we consider simple (not necessarily convex) polygons with n vertices and show that it is possible to find the valid region for cast removal in $O(n^2)$ time. If the aim is to report the valid region, this time bound is worst case optimal.

Let r be a point in the plane. If the valid orientation for cast removal of a simple polygon P changes with respect to r at a reflex vertex $v \in \partial P$, v penetrates $\text{int}(P)$ when rotated infinitesimally around r with arbitrary orientation. This yields the following observation:

Observation 2.3.9. *A rotationally castable polygon P with respect to r cannot be divided at one of its reflex vertices v unless the center of rotation r is v . Hence, v cannot be a far point with respect to r and v can only be a near point with respect to r if $r = v$.*

Definition 2.3.10. *Let v be a vertex of P and denote the two edges adjacent to v by e_1 and e_2 . The near cone of v is defined as $cl(n_{e_1}^-(v) \cap n_{e_2}^-(v))$ and denoted by $NC(v)$.*

The near cone of v is the set of all points $X \in \mathbb{R}^2$ with the property that v is a near point with respect to X , see Figure 2.7.

Definition 2.3.11. *Let v be a vertex of P and denote the two edges adjacent to v by e_1 and e_2 . The far cone of v is defined as $n_{e_1}^+(v) \cap n_{e_2}^+(v)$ and denoted by $FC(v)$.*

The far cone of v is the set of all points $X \in \mathbb{R}^2$ with the property that v is a far point with respect to X , see Figure 2.7.

Definition 2.3.12. *The black region of a reflex vertex v is $(NC(v) \cup FC(v)) \setminus \{v\}$.*

Note that Observation 2.3.9 ensures that the black region of v does not contain any valid centers of rotation r that allow v to be removed from the cast.

Lemma 2.3.13. *For a simple polygon P , $\text{int}(P)$ is contained in the union of the black regions of the edges and the reflex vertices of P .*

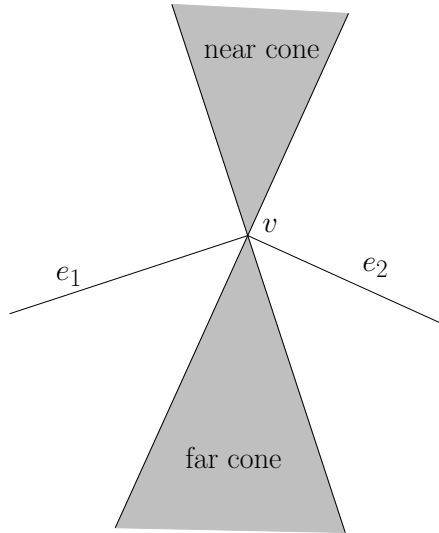


Figure 2.7: *The near cone and the far cone of v .*

Proof. Every point $p \in \text{int}(P)$ has at least one near point $c \in \partial P$ with respect to p . If c is the orthogonal projection of p on the interior of an edge e , p is contained in the black region of e . Otherwise, c is a reflex vertex and p is contained in the black region of c . \square

Lemma 2.3.14. *A simple polygon P is castable with respect to a center of rotation r if and only if r is not contained in the union of all black regions of edges and reflex vertices of P .*

Proof. This proof consists of two parts. First, a simple polygon is not castable with respect to r if r is contained in the union of all black regions of edges and reflex vertices of P . If r is contained in the union of all black regions, it is either contained in the black region of at least one edge e or in the black region of at least one reflex vertex v . Hence, either e or v cannot be removed from the cast.

Second, a simple polygon is always castable if r is not contained in the union of black regions of its edges and reflex vertices. Assume that P is not castable with respect to r and that r is not contained in the union of black regions of edges and reflex vertices of P . Hence, there are at least two far points f_1 and f_2 on ∂P with respect to r , see Theorem 2.2.5 and the Extreme Value Theorem [28, Chapter 3]. Note that

neither f_1 nor f_2 can be a reflex vertex as r is not contained in the black region of any reflex vertex. Two situations are possible: either $r \in \text{int}(P)$ or $r \notin \text{int}(P)$. Lemma 2.3.13 ensures that $r \notin \text{int}(P)$ as any point $q \in \text{int}(P)$ is contained in the union of the black regions of the edges and reflex vertices of P . The following description is illustrated in Figure 2.8. Denote the far point with smallest distance to r by f_1 . If this far point is not unique, choose an arbitrary far point with smallest distance to r . Denote the circle centered at r passing through f_1 by c . In a local neighborhood of f_1 , ∂P is contained in the interior of c . However, since f_2 is a far point on ∂P with respect to r with greater or equal distance from r than f_1 , ∂P intersects c in at least one point not equal to f_1 . Find the first point q_1 of ∂P that intersects c when starting at f_1 and walking along ∂P in ccw orientation. The polygonal chain starting at f_1 and ending at q_1 splits c into two regions. If r is contained in the region of c located to the left of the polygonal chain starting at f_1 and ending at q_1 , we call the polygonal chain an *upper chain*. Otherwise, find the first point q_2 of ∂P that intersects c when starting at f_1 and walking along ∂P in cw orientation. By the Jordan Curve Theorem [32, Chapter 1], the polygonal chain starting at q_2 and ending at f_1 must be completely contained in the region of c located to the left of the polygonal chain starting at f_1 and ending at q_1 . Furthermore, $\text{int}(P)$ is contained in the region bounded by the polygonal chain starting at q_2 and ending at f_1 and by the polygonal chain starting at f_1 and ending at q_1 . Hence, r is contained in the region of c located to the left of the polygonal chain starting at q_2 and ending at f_1 . Denote the polygonal chain starting at q_2 and ending at f_1 by *upper chain*. The points f_1, q_1 , and q_2 are points that maximize the distance from the two polygonal chains considered above to r . Hence, by Observation 2.2.6 there exists a near point c_1 on the upper chain. Since r is located in the region of c located to the left of the upper chain, c_1 cannot be a convex vertex. Hence, c_1 is either located on an edge e of P or c_1 is a reflex vertex of P . If c_1 is located on an edge e , c_1 is the orthogonal projection of r on e and therefore, r is contained in the black region of e . Otherwise, c_1 is a reflex vertex that is a near point and therefore, r is contained in the black region of c_1 . Hence, r is either contained in the black region of the reflex vertex c_1 or in the black region of the edge e . But this contradicts the initial assumption that

r is not contained in the union of all black regions of edges and reflex vertices of P . Hence, P is only castable with respect to r if r is outside of the union of the black regions of the edges and reflex vertices of P . \square

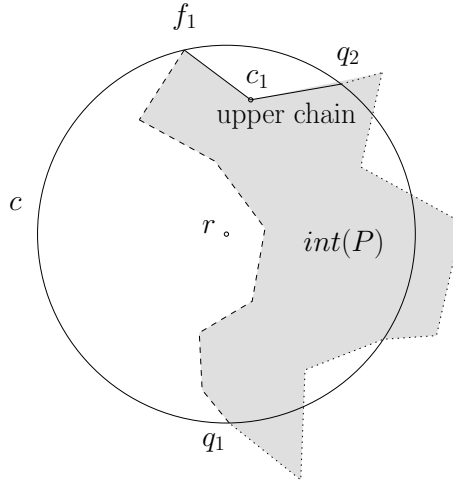


Figure 2.8: The center r is located in the black region of the reflex vertex c_1 .

Theorem 2.3.15. *Given a simple polygon P with n vertices, a description of the valid region for cast removal of P has $O(n^2)$ size and can be computed in $O(n^2)$ time.*

Proof. We preprocess the plane by constructing the full arrangement A of the (full) lines bounding the black regions of edges and reflex vertices. A doubly-connected edge list of the arrangement of n lines has complexity $O(n^2)$ and can be constructed in $O(n^2)$ time. See [21, Chapter 8]. Once A is constructed, each face needs to be labeled as valid or invalid region for cast removal. For this purpose, a boolean value is associated with every edge e and reflex vertex v of P that indicates whether the current location is contained in the black region of e or v respectively. We start at an arbitrary face f of A and test for each edge and reflex vertex of P whether it causes f to be invalid. After testing, we set the boolean value of each edge and reflex vertex appropriately and compute the number b of edges and reflex vertices that cause f to be invalid. Clearly, f is valid if and only if $b = 0$. This computation takes $O(n)$ time as every edge and reflex vertex of P needs to be considered. Next, A is traversed in

depth-first order on the graph induced by the vertices and the edges of A . Each time, an edge e_A of A is crossed, we update both the boolean value of the edge or reflex vertex of P that induces e_A and the counter b . This way, every face of A is labeled in constant time per face. The edge e_A and its incident vertices are valid regions for cast removal if and only if one or more of e_A 's adjacent faces is a valid region for cast removal. Hence, A can be labeled in $O(n^2)$ time. \square

Corollary 2.3.16. *A simple polygon P with n vertices can be preprocessed in $O(n^2)$ time, such that for any query point r , we can decide in $O(\log n)$ time if P is castable with respect to r .*

Proof. Theorem 2.3.15 allows to find the valid region for cast removal of P in $O(n^2)$ time. Hence, the plane is preprocessed, such that every face of the planar subdivision induced by black regions of P is labeled as valid or invalid region in time $O(n^2)$.

For any query point r , after $O(n^2)$ preprocessing time, it is possible to determine the face of the arrangement containing r in time $O(\log n)$ [29]. Once the face is known, the label of the face can be retrieved in constant time. Hence, determining whether r is a valid center of rotation for P takes $O(\log n)$ time. \square

We now examine the complexity of the valid region for centers of rotation. In the best case, i.e. in the case of a convex polygon, the number of valid regions for cast removal is $O(n)$. The number of valid regions cannot be $\omega(n^2)$ as the complexity of an arrangement induced by $O(n)$ lines is $O(n^2)$. Next we show that there exists a class of simple polygons where the number of valid regions is $\Omega(n^2)$. This implies that the $O(n^2)$ time bound is worst case optimal if the aim is to report all valid regions of cast removal for a simple polygon. We now outline the construction of the lower bound.

We construct a simple polygon P consisting of $n = 3s - 1$ vertices located on two different polygonal chains. Let s vertices of P be evenly distributed on the upper half of the unit circle. The coordinates of those vertices are

$$(\cos((i-1)\phi_1), \sin((i-1)\phi_1)), \quad i = 1, \dots, s,$$

where $\phi_1 = \frac{\pi}{s-1}$. Hence, the vertices form a convex polygonal chain c_1 . All valid regions induced by c_1 are cones whose apexes a are on the unit circle and opening angle $\frac{\phi_1}{2}$, see Figure 2.9.

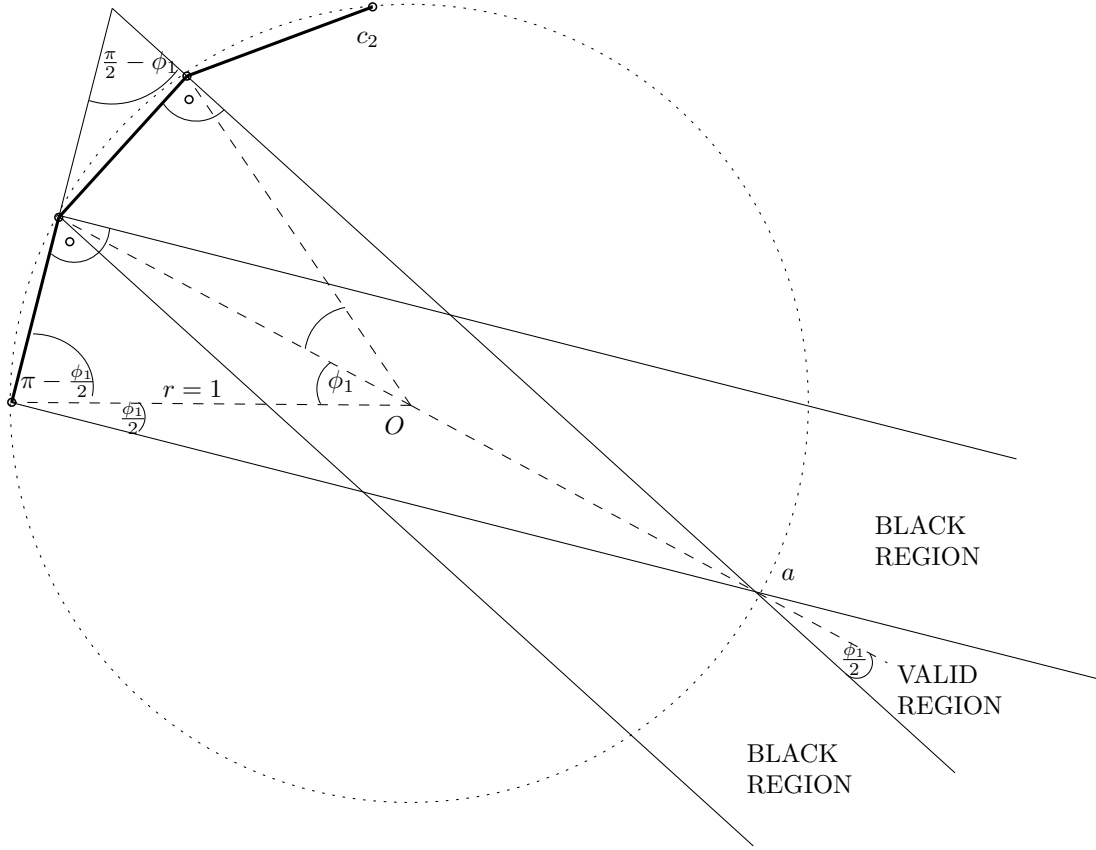


Figure 2.9: *Approximation of half circle*

The second polygonal chain c_2 consists of $2s - 1$ vertices. Let s vertices of c_2 be evenly distributed on the arc of the circle with center $(-\frac{1}{2}, 0)$ and radius 1 starting at $\frac{3\pi}{2}$ and ending at $\frac{25\pi}{16}$. The coordinates of those vertices are

$$\left(-\frac{1}{2} + \cos\left(\frac{3\pi}{2} + (i-1)\phi_2\right), \sin\left(\frac{3\pi}{2} + (i-1)\phi_2\right) \right), \quad i = 1, \dots, s,$$

where $\phi_2 = \frac{\pi}{16(s-1)}$. Denote the vertices by v_1, \dots, v_s and note that v_i is not located in the interior of the unit disk for $i = 1, \dots, s$. Define the vertices v_0, v_{s+1} as

$$\left(-\frac{1}{2} + \cos\left(\frac{3\pi}{2} - \phi_2\right), \sin\left(\frac{3\pi}{2} - \phi_2\right) \right), \left(-\frac{1}{2} + \cos\left(\frac{3\pi}{2} + s\phi_2\right), \sin\left(\frac{3\pi}{2} + s\phi_2\right) \right),$$

respectively. Let $s - 1$ vertices of c_2 be defined as the intersections of the line passing through v_{i-1} and v_i with the line passing through v_{i+1} and v_{i+2} , where $i = 1, \dots, s - 1$. These vertices are located on a circle. The polygonal chain c_2 consists of $s - 2$ reflex, $s - 1$ convex, and 2 boundary vertices. Note that c_2 consists of sides of isosceles triangles, i.e. all the edges have the same length, see Figure 2.10. Valid regions bounded by part of c_2 and two parallel half lines occur.

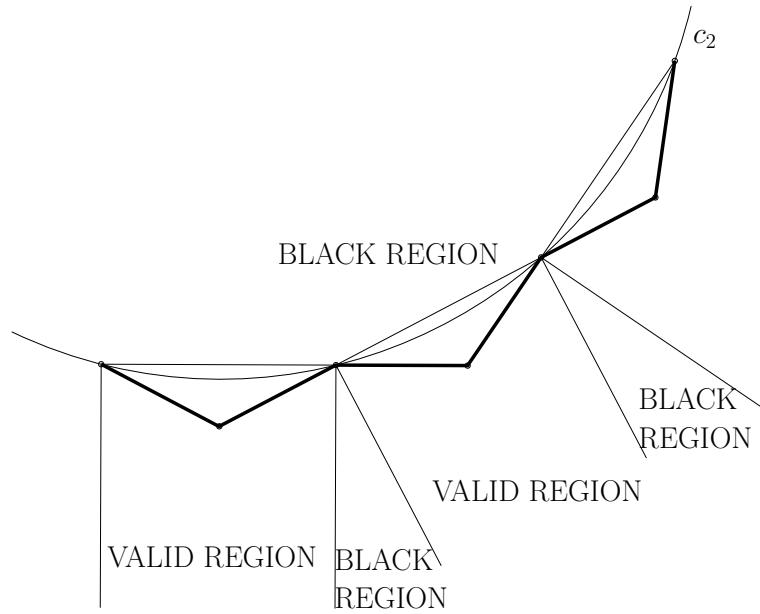


Figure 2.10: *Polygonal chain*

The two polygonal chains c_1 and c_2 can now be connected by two edges. This does not introduce further reflex vertices to P , but only two black regions of the new edges. Those black regions have no influence on further considerations. Each of the black regions induced by reflex vertices on c_2 induces a bounded valid region when intersecting the valid region induced by vertices located on the arc of c_1 starting at $\frac{7\pi}{16}$ and ending at $\frac{\pi}{2}$. Hence, there are at least $(s - 2) \lfloor \frac{s}{8} \rfloor$ bounded valid regions. As $n = 3s - 1$, there are $\frac{n-5}{3} \lfloor \frac{n+1}{24} \rfloor = \Omega(n^2)$ bounded valid regions. Hence, the number of valid regions for cast removal of a simple polygon is $\Omega(n^2)$. An example with $s = 10$ is shown in Figure 2.11.

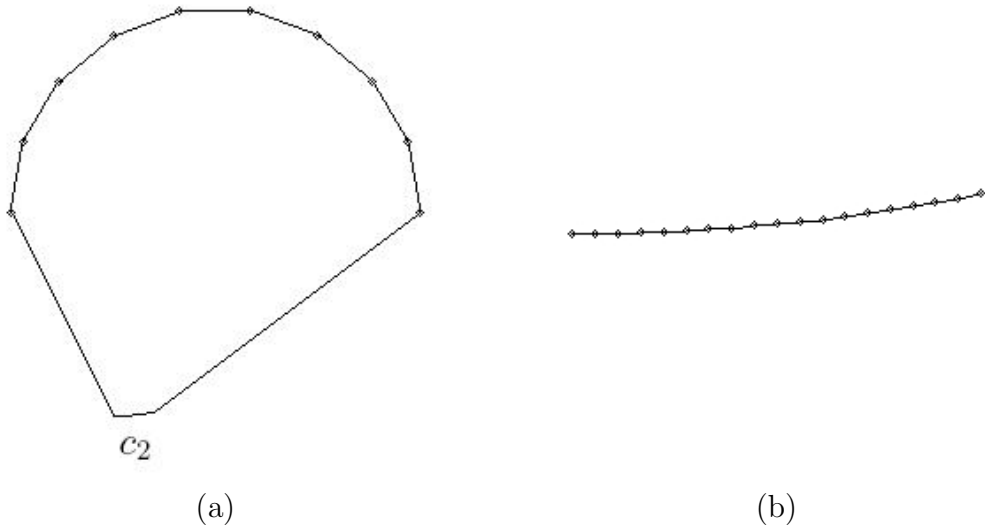


Figure 2.11: *Example with $s = 10$. (a) shows the polygon, (b) shows an enlargement of the polygonal chain c_2 .*

Chapter 3

Rotational Casting in Three Dimensions

Every dimension is special.

JEFF LAGARIAS

In this chapter, the results of the previous chapter are extended to solve rotational castability of 3-dimensional objects modeled by polyhedra. Although the conceptual approach used in this chapter is the same as in two dimensions, the methods to solve the problem are more complex.

Section 3.1 introduces the notation and preliminaries used throughout the chapter, Section 3.2 discusses the problem of finding a partitioning of the cast based on a given line of rotation, and Section 3.3 discusses the problem of finding all of the lines in space that allow a valid partitioning of the cast.

3.1 Preliminaries

Define a polyhedron P of arbitrary genus as a closed, compact, connected subset of \mathbb{R}^3 bounded by a finite number of piecewise linear surface. Let $int(P)$ and ∂P denote the interior and boundary of P , respectively, so that $P = int(P) \cup \partial P$. The boundary is also called the *cast* of P . Two faces are *adjacent* if they share at least one edge.

Parallel adjacent faces are not allowed, since this can be easily avoided by merging pairs of adjacent parallel faces. Let n denote the combinatorial complexity of P^1 . As in the 2-dimensional problem, the aim is to rotationally remove the cast of P in two pieces.

Definition 3.1.1. *Let l be a directed line in 3-dimensional space. Consider the plane π perpendicular to l passing through a point p of P . Denote the intersection point of π and l by $l'(p)$. A face f of P is called removable in cw orientation with respect to l if*

$$\exists \alpha > 0 \text{ such that } \forall p \in \text{int}(f) : \text{cwarc}(l'(p), p, \alpha) \cap \text{int}(P) = \emptyset$$

and removable in ccw orientation with respect to l if

$$\exists \alpha > 0 \text{ such that } \forall p \in \text{int}(f) : \text{ccwarc}(l'(p), p, \alpha) \cap \text{int}(P) = \emptyset.$$

The cw and ccw orientation is measured with respect to the orientation of line l .

The cw or ccw orientation is then called a *valid orientation for cast removal for f with respect to l* respectively, and l is called a *valid casting line for f* .

Definition 3.1.2. *Let l be a directed line in 3-dimensional space. A polyhedron P is rotationally castable with respect to l , if ∂P can be partitioned into exactly two non-empty connected subsets P_1 and P_2 , such that all faces of P_1 are removable in cw orientation with respect to l , all faces of P_2 are removable in ccw orientation with respect to l ,*

$$\exists \alpha > 0 \text{ such that } \forall p \text{ on } P_1 : \text{cwarc}(l'(p), p, \alpha) \cap \text{int}(P_2) = \emptyset,$$

and

$$\exists \alpha > 0 \text{ such that } \forall p \text{ on } P_2 : \text{ccwarc}(l'(p), p, \alpha) \cap \text{int}(P_1) = \emptyset.$$

The last two conditions of the definition ensure that the two subsets P_1 and P_2 (which partition the boundary of P) do not obstruct each other's rotational paths. Figure 3.1 shows part of a polyhedron that is decomposed into P_1 and P_2 . Although

¹Let E , F , and V denote the number of edges, faces, and vertices of P respectively. Then $n = E + F + V$.

∂P can be decomposed into P_1 and P_2 , such that P_1 is removable in cw orientation and P_2 is removable in ccw orientation, the two components cannot be rotated infinitesimally, because they obstruct each other. In the following, the notations rotationally castable and castable are used interchangeably. Note that the partitioning of the polyhedron is not necessarily at edges of P .

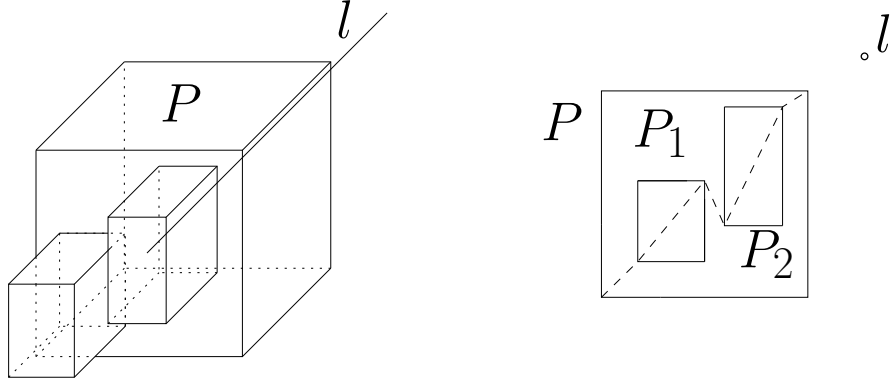


Figure 3.1: Two different views of P are shown along with the partitioning of P . The components P_1 and P_2 obstruct each other's rotational paths.

Let $p \in \partial P$ be a point incident to k reflex edges $e_i, 1 \leq i \leq k$ and s faces $f_i, 1 \leq i \leq s$. Denote the direction of e_i by $\vec{d}(e_i)$ and the inner normal of f_i by $\vec{n}(f_i)$.

Definition 3.1.3. Let l be a directed line in 3-dimensional space in direction \vec{l} . A polyhedron P is robustly castable with respect to l , if P is rotationally castable with respect to l ,

$$\forall p \in \partial P : \vec{d}(e_i) \cdot \vec{l} \neq 0, \quad \forall 1 \leq i \leq k,$$

and

$$\forall p \in \partial P : \vec{n}(f_i) \times \vec{l} \neq \mathbf{0}, \quad \forall 1 \leq j \leq s,$$

where \cdot denotes the dot product of two vectors and \times denotes the cross product of two vectors.

The last two conditions of the definition ensure that no face or reflex edge of P is perpendicular to l .

If P is robustly castable with respect to l , we say l is a *valid robust casting line* for P . The definition implies that P is robustly castable with respect to a line l if

P is castable with respect to l and no reflex edge or face of P is perpendicular to l . However, there can still exist a point of ∂P that slides along the boundary of P when the cast parts are removed. This is the case when the orthogonal projection of l on a removable face of P splits the face into two parts, see Lemma 3.2.2. Hence, the definition does not correspond to the intuitive definition of robust, where the limited precision of the casting machinery used to manufacture an object is taken into account. Therefore, the definition of robust castability does not imply that surface defects are less likely to occur in robust casting than in general casting. Although the definition of robust castability does not offer this advantage, it ensures that algorithms to determine robust castability of a polyhedron are more efficient than algorithms to determine general castability of a polyhedron.

The following sections discuss both the robust casting process and the general casting process.

3.2 Decision Problem

In this section, we address the problem of determining whether a polyhedron P with combinatorial complexity n is castable with respect to a given line of rotation and present an algorithm that solves the problem in $O(n \log n)$ time. In case we want to determine if P is robustly castable, the algorithm's running time becomes $O(n)$.

Let P be a polyhedron and let l be a line in space. The problem can be decomposed into three subproblems: determining the valid orientation for cast removal for all faces of P with respect to l , checking whether all the faces removable in cw and ccw orientation with respect to l form a connected set, and testing whether the two components can be rotated by a small amount around l without colliding.

3.2.1 Robust rotational casting

This section presents an algorithm to test whether a polyhedron P is robustly castable with respect to a line l in space in $O(n)$ time.

Determining the valid orientation of a face

In this section, we solve the problem of determining the valid orientation with respect to a line l of every face of a polyhedron P with combinatorial complexity n in $O(n)$ time. Denote the position vector of a point p by \vec{p} .

Definition 3.2.1. Let f be a face of P and denote its inner normal by \vec{n} . Define the open unbounded prism $S(f) = \{\vec{p} + t\vec{n} : p \in f \text{ and } t \in \mathbb{R}\}$. Denote $S^+(f) = \{\vec{p} + t\vec{n} : p \in f \text{ and } t > 0\}$ and $S^-(f) = \{\vec{p} + t\vec{n} : p \in f \text{ and } t \leq 0\}$, see Figure 3.2.

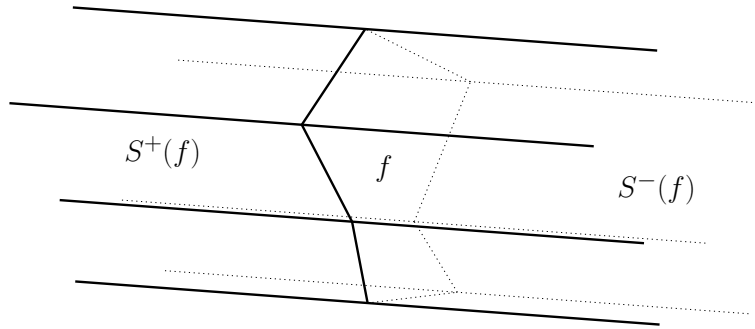


Figure 3.2: The prisms $S^+(f)$ and $S^-(f)$.

In a first step, the algorithm determines whether any reflex edge or face of P is perpendicular to the given line l . This test takes constant time per reflex edge or face, respectively, and therefore $O(n)$ total time. If any reflex edge or face of P is perpendicular to l , P is not robustly castable with respect to l , by definition. Otherwise, we determine the valid orientation for cast removal for each face of P . In the following, we can therefore assume that no reflex edge or face of P is perpendicular to l .

The following lemma, which is a 3-dimensional version of Lemma 2.1.3, characterizes all locations from which a face is removable.

Let f be a face of P with inner normal \vec{n} and let l be a line in direction \vec{l} with $\vec{n} \times \vec{l} \neq \mathbf{0}$.

If $l \cap S(f) = \emptyset$, compute a point $p \in \partial f$ that minimizes the Euclidean distance between ∂f and l . If p is not unique, the set of points on ∂f with minimal distance from l are located on an edge of ∂f parallel to l and an arbitrary point p of that set

is picked. Denote by τ the plane with normal $\vec{n} \times \vec{l}$ passing through p and by p^* the point p translated by $\vec{n} \times \vec{l}$. Denote the open half space induced by τ containing p^* by τ^+ and the open half space induced by τ not containing p^* by τ^- .

Lemma 3.2.2. *For the orientation for cast removal of f , the following four cases are possible:*

1. *The face f is removable from the cast using only a cw orientation around l , if and only if $l \cap S(f) = \emptyset$ and $l \in cl(\tau^-)$.*
2. *The face f is removable from the cast using only a ccw orientation around l , if and only if $l \cap S(f) = \emptyset$ and $l \in cl(\tau^+)$.*
3. *The face f needs to be partitioned into two or more parts along the orthogonal projection of l on f in order to be removed, if and only if $l \cap S^+(f) = \emptyset$ and $l \cap S^-(f) \neq \emptyset$. One or more parts of f are removable using a ccw rotation and the other ones using a cw rotation around l .*
4. *The face f is not removable from the cast, if and only if $l \cap S^+(f) \neq \emptyset$.*

Proof. Consider the plane π perpendicular to l passing through a point p of P . Denote the intersection point of π and l by $l'(p)$. Every point $p \in int(f)$ moves on $cwarc(l'(p), p, \alpha)$ or $ccwarc(l'(p), p, \alpha)$ when rotated by an angle α around l . As in the proof of Lemma 2.1.3, denote the vector from p to $l'(p)$ by $p\vec{l}'(p)$ and the vector $\vec{l} \times p\vec{l}'(p)$ by $p\vec{l}'(p)^\perp$. The tangent of $cwarc(l'(p), p, \alpha)$ or $ccwarc(l'(p), p, \alpha)$ is $p\vec{l}'(p)^\perp$ or $-p\vec{l}'(p)^\perp$, respectively, for any point p . Denote by ν the plane passing through p with normal vector $p\vec{l}'(p)$ and denote by ν^+ the open half space bounded by ν containing l' . The two arcs $cwarc(l'(p), p, \alpha)$ and $ccwarc(l'(p), p, \alpha)$ are contained in $cl(\nu^+)$.

Let p be an arbitrary point in the interior of f . There exists an open ball b with positive radius centered at p with the property that exactly half of b is contained in $int(P)$ and exactly half of b is contained in the exterior of P . Denote the ray starting at p propagating in direction $p\vec{l}'(p)^\perp$ by q^+ and denote the ray starting at p propagating in direction $-p\vec{l}'(p)^\perp$ by q^- .

Assume that $l \cap S(f) = \emptyset$ and $l \in cl(\tau^-)$, and let p be an arbitrary point in the interior of f . The intersection $b \cap q^+$ is located completely outside of $int(P)$. Hence, p can move by a small amount along $\vec{p}l'(p)^\perp$ without penetrating $int(P)$. Since $cwarc(l'(p), p, \alpha) \subseteq cl(\nu^+)$ and since $\vec{p}l'(p)^\perp$ is the tangent of $cwarc(r, p, \alpha)$ in p , small movements of p along $cwarc(l'(p), p, \alpha)$ are possible without penetrating $int(P)$. Hence, $\exists \alpha > 0$ such that $\forall p \in int(f) : cwarc(l'(p), p, \alpha) \cap int(P) = \emptyset$. The intersection $b \cap q^-$ is completely contained in $int(P) \cup \{p\}$ and hence, p cannot move infinitesimally along $-\vec{p}l'(p)^\perp$ without penetrating $int(P)$. Since infinitesimal movements along $-\vec{p}l'(p)^\perp$ correspond to infinitesimal movements along $ccwarc(l'(p), p, \alpha)$, there is no $\alpha > 0$ such that $\forall p \in int(f) : ccwarc(l'(p), p, \alpha) \cap int(P) = \emptyset$. Hence, f is only removable using a cw orientation around l .

Assume that $l \cap S(f) = \emptyset$ and $l \in cl(\tau^+)$, and let p be an arbitrary point in the interior of f . The intersection $b \cap q^-$ is located completely outside of $int(P)$. Hence, p can move by a small amount along $-\vec{p}l'(p)^\perp$ without penetrating $int(P)$. Since $ccwarc(l'(p), p, \alpha) \subseteq g^+(p)$ and since $-\vec{p}l'(p)^\perp$ is the tangent of $ccwarc(l'(p), p, \alpha)$ in p , small movements of p along $ccwarc(l'(p), p, \alpha)$ are possible without penetrating $int(P)$. Hence, $\exists \alpha > 0$ such that $\forall p \in int(f) : ccwarc(l'(p), p, \alpha) \cap int(P) = \emptyset$. The intersection $b \cap q^+$ is completely contained in $int(P) \cup \{p\}$ and hence, p cannot move infinitesimally along $\vec{p}l'(p)^\perp$ without penetrating $int(P)$. Since infinitesimal movements along $\vec{p}l'(p)^\perp$ correspond to infinitesimal movements along $cwarc(l'(p), p, \alpha)$, there is no $\alpha > 0$ such that $\forall p \in int(f) : cwarc(l'(p), p, \alpha) \cap int(P) = \emptyset$. Hence, f is only removable using a ccw orientation around l .

If $l \cap S^+(f) = \emptyset$ and $l \cap S^-(f) \neq \emptyset$, f is divided into two or more faces along the perpendicular projection l' of l on f by inserting one or more edges e . Denote the set of faces contained in $cl(\tau^-)$ by F_1 and the set of faces contained in $cl(\tau^+)$ by F_2 . For arbitrary $f_1 \in F_1$ and $f_2 \in F_2$, the points on e globally minimize the distance between $\partial f_1, \partial f_2$ and l respectively. Hence, f_1 is removable using a cw orientation around l and f_2 is removable using a ccw orientation around l . Note that F_1 and F_2 contain exactly one face each for every convex face f .

If $l \cap S^+(f) \neq \emptyset$, consider an arbitrary point p located on the orthogonal projection of l on f . The point p cannot be rotationally removed from the cast, i.e. there is no

$\alpha > 0$ such that $cwarc(l'(p), p, \alpha) \cap \text{int}(P) = \emptyset$ or $ccwarc(l'(p), p, \alpha) \cap \text{int}(P) = \emptyset$. Therefore, f is not removable with respect to l .

This determines the removability of f depending on the location of l . Hence, the four statements of Lemma 3.2.2 follow directly. \square

Lemma 3.2.2 classifies all faces of P into four categories. In linear time, all the faces of P can be categorized. If any face of P is not removable with respect to l , P is not castable with respect to l . Otherwise, the aim is to determine if the boundary of P can be decomposed into two connected sets. In particular, the faces of P that need to be split (faces of type 3 in Lemma 3.2.2) need to be examined more carefully as outlined in the following lemma.

Lemma 3.2.3. *Let P be a polyhedron with combinatorial complexity n and let l be a line in space. All the faces f of P with $l \cap S^+(f) = \emptyset$ and $l \cap S^-(f) \neq \emptyset$ can be partitioned along the orthogonal projection of l on f and the resulting new faces can be labeled as removable in cw or ccw orientation with respect to l respectively in $O(n)$ time.*

Proof. Consider a single face f with $l \cap S^+(f) = \emptyset$ and $l \cap S^-(f) \neq \emptyset$. Lemma 3.2.2 states that f needs to be partitioned along the orthogonal projection of l on f . If we denote the number of vertices of f by r , f is partitioned into $O(r)$ faces, because every edge of f is split at most once. Assume that f is stored using a doubly-connected edge list. The $s \leq r$ points on ∂f , where f needs to be split can be found ordered by their appearance on f in $O(r)$ time by traversing the edges of f [21, Chapter 2]. Using the algorithm by Hoffmann et al. [25], it is possible to order the points with respect to their appearance on the projected line in time $O(r)$. Denote the points on ∂f , where f needs to be split ordered with respect to their appearance on the projected line by q_1, \dots, q_s . Note that s is even, because every edge that is added to f has two endpoints and no two new edges share a vertex. Adding the edges with endpoints q_1 and q_2 , q_3 and q_4 , \dots , q_{s-1} and q_s partitions f along the orthogonal projection of f on l . Once the new edges and their intersection points with f are known, it takes time $O(m)$ to insert the new edge into the doubly-connected edge

list, where m is the number of edges adjacent to the new edge [21, Chapter 2]. In the worst case, $O(r) = O(n)$ and $O(m) = O(n)$.

Next, we analyze the running time for splitting all of the faces f with $l \cap S^+(f) = \emptyset$ and $l \cap S^-(f) \neq \emptyset$ of P . Since every edge e of P has two adjacent faces, e needs to be split at most twice. This ensures that the total number of new edges is $O(n)$. Also, the total number of edges adjacent to the new edges is $2n = O(n)$. The reason for this is that each original edge of P is adjacent to at most two new edges. Hence, all of the faces can be split in $O(n)$ total time. Each of the newly created faces can be labeled as removable in cw or ccw orientation with respect to l , respectively, in constant time. Therefore, it takes $O(n)$ time to partition all the faces of P with $l \cap S^+(f) = \emptyset$ and $l \cap S^-(f) \neq \emptyset$ and to find the valid orientation with respect to l for each of the newly created faces. \square

This yields the conclusion that all the faces of a polyhedron P with combinatorial complexity n can be labeled as removable in cw, ccw, or no orientation with respect to a line of rotation l in $O(n)$ total time.

Testing Connectivity

Given a polyhedron P and a line l in space, all the faces of P can be labeled as removable in cw, ccw, or no orientation using the methods presented in the previous section. If any of P 's faces is not removable with respect to l , P is not robustly castable with respect to l . Otherwise, it is crucial to test whether all the faces removable in cw or ccw orientation with respect to l form connected sets, respectively. The following lemma allows this test in linear time:

Lemma 3.2.4. *Let P be a polyhedron and let l be a line in space. Given a partitioning of ∂P into two sets P_1 and P_2 , such that all faces of P_1 are removable in cw orientation with respect to l and all faces of P_2 are removable in ccw orientation with respect to l , we can determine whether P_1 and P_2 are connected respectively in time $O(n)$.*

Proof. To test whether all the faces of P_1 and P_2 form connected sets, consider two undirected graphs: graph G_1 induced by the faces in P_1 and graph G_2 induced by

the faces in P_2 . Next, a depth-first search [18, Chapter 22] is performed on G_1 and G_2 respectively. The faces in P_1 and P_2 are connected, respectively, if and only if the depth-first forests of G_1 and G_2 consist of exactly one tree each. This implies that non-connected polyhedra are not castable with respect to any line. The running time for this algorithm is in the order of the sum of all the edges and vertices of the two undirected graphs, and hence $O(n)$. \square

Testing Collisions

The results presented in the two previous sections allow to test whether a polyhedron P can be partitioned into two connected sets P_1 and P_2 , such that all faces of P_1 are removable in cw orientation with respect to l and all faces of P_2 are removable in ccw orientation with respect to l . If ∂P cannot be partitioned this way then P is not robustly castable with respect to l . Otherwise, it remains to test whether P_1 can be rotated by an arbitrarily small angle in cw orientation with respect to l without colliding with P_2 and whether P_2 can be rotated by an arbitrarily small angle in ccw orientation with respect to l without colliding with P_1 . The definition of robust castability of P implies that P is robustly castable if and only if the rotations are possible.

Lemma 3.2.5. *Let P be a polyhedron and let l be a line in space. Given a partitioning of ∂P into two connected sets P_1 and P_2 , such that all faces of P_1 are removable in cw orientation with respect to l and all faces of P_2 are removable in ccw orientation with respect to l , we can test whether*

$$\exists \alpha > 0 \text{ such that } \forall q \text{ on } P_1 : \text{cwar}c(l'(q), q, \alpha) \cap \text{int}(P_2) = \emptyset$$

or equivalently

$$\exists \alpha > 0 \text{ such that } \forall q \text{ on } P_2 : \text{ccw}ar\text{e}(l'(q), q, \alpha) \cap \text{int}(P_1) = \emptyset$$

in time $O(n)$.

Proof. Each edge contained in the set of cycles C consisting of edges and vertices of P that separates P_1 from P_2 can be found in linear time by testing for each edge

of P whether it is adjacent to both a face removable in cw orientation with respect to l and a face removable in ccw orientation with respect to l . The statement of Lemma 3.2.5 is true if and only if every point in C can rotate by an arbitrarily small angle in cw (respectively ccw) orientation with respect to l without penetrating P_2 (respectively P_1). As C consists of non intersecting simple polygonal cycles on P , it is sufficient to test all the vertices of C . Denote the intersection of the plane π perpendicular to l passing through a vertex p of C with l by $l'(p)$. Rotations of p with respect to l in 3-dimensional space correspond to rotations of p with respect to $l'(p)$ in π . Lemma 2.2.4 of Chapter 2 ensures that p can be rotated by an arbitrarily small angle in arbitrary orientation with respect to $l'(p)$ if and only if p is a near or a far point with respect to $l'(p)$ and p is not a reflex vertex in π . Hence, C can rotate by an arbitrarily small angle in cw (respectively ccw) orientation with respect to l without penetrating P_2 (respectively P_1), if every vertex p of C is a near or far point with respect to $l'(p)$ and is not a reflex vertex in π .

The aim is to test for every point p of C whether it either locally minimizes or maximizes the distance between $\partial P \cap \pi$ and $l'(p)$ and is not a reflex vertex in π . Note that p is incident to two edges in C , because C is a set of non-intersecting simply closed cycles on P . Hence, testing whether p is a local extremum requires intersecting π with the four faces adjacent to the two edges of C adjacent to p . We can test whether p minimizes or maximizes the distance to $l'(p)$ by computing the distances from p and p 's two neighboring vertices in π to $l'(p)$. If p is closer to $l'(p)$ than its two neighboring vertices, it locally minimizes the distance between $\partial P \cap \pi$ and $l'(p)$. If p is further from $l'(p)$ than its two neighboring vertices, it locally maximizes the distance between $\partial P \cap \pi$ and $l'(p)$. Hence, testing whether one vertex p of C can rotate by an arbitrarily small angle in cw (respectively ccw) orientation with respect to l without penetrating $int(P_2)$ (respectively $int(P_1)$) takes constant time. As C has $O(n)$ vertices, testing whether P_1 and P_2 collide takes $O(n)$ time. \square

This yields the following result:

Theorem 3.2.6. *Given a polyhedron P with combinatorial complexity n and a line of rotation l in 3-dimensional space, it is possible to determine whether P is robustly*

castable with respect to l in time $O(n)$.

3.2.2 General rotational casting

This section presents an algorithm to test whether a polyhedron P is castable with respect to a line l in space in $O(n \log n)$ time.

Determining the valid orientation of a face

In this section, we solve the problem of determining the valid orientation with respect to a line l of every face of a polyhedron P with combinatorial complexity n in $O(n \log n)$ time.

Consider the notation of Lemma 3.2.2 in Section 3.2.1. In the case of general castability of P , it is possible that a face f of P with inner normal \vec{n} is perpendicular to the given line l .

Observation 3.2.7. *Let f be a face of P with inner normal \vec{n} and let l be a line in direction \vec{l} . If $l \cap S(f) = \emptyset$ and $\vec{n} \times \vec{l} = \mathbf{0}$, f is removable in both cw and ccw orientation with respect to l .*

Proof. The idea of the proof is similar to the proof of Lemma 3.2.2. Consider the supporting plane π of f . Denote the intersection point of π and l by $l'(p)$. Every point p of f moves on $cwarc(l'(p), p, \alpha)$ or $ccwarc(l'(p), p, \alpha)$ when rotated by an angle α around l . As in the proof of Lemma 3.2.2, denote the vector from p to $l'(p)$ by $\vec{pl}'(p)$ and the vector $\vec{l} \times \vec{pl}'(p)$ by $\vec{pl}'(p)^\perp$. The tangent of $cwarc(l'(p), p, \alpha)$ or $ccwarc(l'(p), p, \alpha)$ is $\vec{pl}'(p)^\perp$ or $-\vec{pl}'(p)^\perp$, respectively, for any point p . Denote by ν the plane passing through p with normal vector $\vec{pl}'(p)$ and denote by ν^+ the open half space bounded by ν containing l' . The two arcs $cwarc(l'(p), p, \alpha)$ and $ccwarc(l'(p), p, \alpha)$ are contained in $cl(\nu^+)$.

Let p be an arbitrary point in the interior of f . There exists an open ball b with positive radius centered at p with the property that exactly half of b is contained in $int(P)$ and exactly half of b is contained in the exterior of P . Denote the ray starting at p propagating in direction $\vec{pl}'(p)^\perp$ by q^+ and denote the ray starting at p propagating in direction $-\vec{pl}'(p)^\perp$ by q^- .

Let $l \cap S(f) = \emptyset$, let \vec{l} be parallel to \vec{n} , and let p be an arbitrary point in the interior of f . Both the intersection $b \cap q^+$ and the intersection $b \cap q^-$ are a subset of ∂P and hence, completely outside of $\text{int}(P)$. Hence, p can move by a small amount along both $p\vec{l}'(p)^\perp$ and $-p\vec{l}'(p)^\perp$ without penetrating $\text{int}(P)$. Therefore, $\exists \alpha > 0$ such that $\forall p$ on $f : \text{cwarc}(l'(p), p, \alpha) \cap \text{int}(P) = \emptyset$ and $\exists \alpha > 0$ such that $\forall p$ on $f : \text{ccwarc}(l'(p), p, \alpha) \cap \text{int}(P) = \emptyset$. Hence, f is removable using both a cw and a ccw rotation around l . \square

As in Section 3.2.1, it is possible to label all the faces of P that are not perpendicular to l as removable in cw, ccw, or no orientation in $O(n)$ time. It remains to split the faces with $l \cap S(f) = \emptyset$ and $\vec{n} \times \vec{l} = \mathbf{0}$ and assign unique valid orientations to the newly created faces if possible. The face f can be partitioned into two (possibly empty) sets of faces F_1 and F_2 , such that all the faces adjacent to f removable in cw orientation with respect to l are adjacent to faces in F_1 and all the faces adjacent to f removable in ccw orientation with respect to l are adjacent to faces in F_2 . The definition of castability of a polyhedron implies that f is only removable with respect to l if there exists a partitioning of f into F_1 and F_2 with the following two properties: F_1 does not intersect F_2 when rotated by an arbitrarily small angle in cw orientation with respect to l and F_2 does not intersect F_1 when rotated by an arbitrarily small angle in ccw orientation with respect to l . Such a partitioning is called a *valid partitioning of f* . The definition of castability implies that P is not castable with respect to l if no valid partitioning of f exists. Then, we say that f is not removable with respect to l .

If l passes through a hole of a face f perpendicular to l , there are two possible cases: $l \cap \partial P \neq \emptyset$ and $l \cap \partial P = \emptyset$. If $l \cap \partial P \neq \emptyset$, at least one of the faces of P intersects l and is therefore not removable with respect to l . If $l \cap \partial P = \emptyset$, the boundary of the hole of P that contains l cannot be removed without penetrating $\text{int}(P)$. This yields the following observation:

Observation 3.2.8. *Let P be a polyhedron with combinatorial complexity n , let f be a face of P and let l be a line perpendicular to f . If l passes through a hole of f , P is not castable with respect to l .*

Lemma 3.2.9. *Let P be a polyhedron with combinatorial complexity n and let l be a line in direction \vec{l} . For all faces f of P with inner normal \vec{n} , $l \cap S(f) = \emptyset$, and \vec{l} is parallel to \vec{n} , we can find a valid partitioning or report that f is not removable with respect to l in total time $O(n \log n)$.*

Proof. Consider a single face f with inner normal \vec{n} , $l \cap S(f) = \emptyset$, and \vec{l} is parallel to \vec{n} . As f is located in a plane orthogonal to l and as no two adjacent faces of P are parallel, we know the valid orientation for cast removal with respect to l for every face adjacent to f . An edge of f is considered removable in cw (respectively ccw) orientation with respect to l if it is incident to a face adjacent to f removable in cw (respectively ccw) orientation with respect to l . Since the valid orientations with respect to l of all the faces adjacent to f are known, every edge of f can be labeled as removable in cw or ccw orientation with respect to l , respectively, in constant time per edge. The aim is to partition f into two or more components F_1 and F_2 , such that all the edges of f removable in cw orientation with respect to l are only incident to F_1 and all the edges removable in ccw orientation with respect to l are only incident to F_2 .

Denote the set of edges of f removable in cw orientation with respect to l by $CW(f)$ and the set of edges of f removable in ccw orientation with respect to l by $CCW(f)$. Define $cw\alpha_1(p) \geq 0$ as the maximal angle by which p can be rotated in cw orientation with respect to l without penetrating the exterior of f . Let $cw\alpha_2 > 0$ be an arbitrarily small positive angle. Define $cw\alpha(p) = \max(cw\alpha_1(p), cw\alpha_2)$ and note that $cw\alpha(p) > 0$. Let l' denote the point $l \cap \pi$, where π is the supporting plane of f .

Claim 3.2.10. *There exists a valid partitioning of f if and only if*

$$(cwarcc(l', p, cw\alpha(p)) \setminus \{p\}) \cap CCW(f) = \emptyset \quad \forall p \in CW(f).$$

The proof of this claim is left until later.

Next, we present an algorithm to decide whether $(cwarcc(l', p, cw\alpha(p)) \setminus \{p\}) \cap CCW(f) = \emptyset \quad \forall p \in CW(f)$ for a given face f and analyze its running time. Each edge of f is labeled as removable in cw or ccw orientation with respect to l respectively.

The aim is to test whether any point p on an edge removable in cw orientation with respect to l intersects an edge removable in ccw orientation with respect to l when rotated by $cw\alpha(p)$ in cw orientation with respect to l . To conduct this test, f is represented in a fixed polar coordinate system, where l' is the origin and where the polar axis is not perpendicular to any edge of f and does not intersect f . If it is impossible to represent f that way, then P is not castable with respect to l , because the initial assumption ensures that $l' \notin \text{int}(f)$ and Observation 3.2.8 shows that P is not castable with respect to l if l passes through a hole of f . In polar coordinates, every point is described by an angle ϕ and the distance d from the origin. Transform the coordinate system into a cartesian coordinate system, such that the x -axis describes the angle ϕ and the y -axis describes the distance d . Considering this transformed system between $x = 0$ and $x = 2\pi$ shows the transformed polar coordinate system. The polygon f is transformed into \tilde{f} . Since the positive polar axis does not intersect f , f and \tilde{f} are topologically equivalent. The vertices of \tilde{f} are points and the edges of \tilde{f} are transformed line segments. The transformation of a line l is in essence a curve describing the distance of points on the line to the origin. The distance of points on l to the origin has exactly one minimum and consists of two strictly monotone pieces with monotone derivatives. Hence, there exists a pair (d_0, ϕ_0) that denotes the minimum distance from the line to the origin and the distance increases symmetrically as ϕ increases or decreases, respectively. This yields that edges of \tilde{f} have one global minimum and consist of at most two strictly monotone pieces. Furthermore, each edge e of f was split before (while splitting faces of type 4 in Lemma 3.2.2) at the perpendicular projection of l' on e . This corresponds to splitting each edge \tilde{e} of \tilde{f} at the global minimum. Hence, every edge of \tilde{f} is a strictly monotone curve with monotone derivative. Two edges of \tilde{f} only intersect in vertices of \tilde{f} . Each edge \tilde{e} has constant description size.

In the cartesian coordinate system, rotations in cw (respectively ccw) orientation with respect to l' correspond to translations in direction of the negative (respectively positive) x -axis. It is possible to determine whether f is removable with respect to l by enlarging a circle in the polar coordinate system. This corresponds to sweeping a line in y -direction in the cartesian plane. Let each edge \tilde{e} of \tilde{f} be labeled as removable

in cw or ccw orientation with respect to l respectively and let \tilde{e} have a label indicating whether $int(\tilde{f})$ is located to the left or to the right side of \tilde{e} . Let the sweep line status of the plane sweep algorithm contain the edges of \tilde{f} that are currently intersected by the sweep line and let the event queue contain all the vertices of \tilde{f} . At every event point of the plane sweep algorithm, an edge of \tilde{f} is either inserted to or removed from the sweep line status. If this insertion or removal, respectively, yields one of the following two situations, the algorithm reports that f is not removable with respect to l .

1. An edge \tilde{e}_1 removable in cw orientation with respect to l is located immediately to the right of an edge \tilde{e}_2 removable in ccw orientation with respect to l and $int(\tilde{f})$ is located to the left of \tilde{e}_1 .
2. An edge \tilde{e}_1 removable in cw orientation with respect to l is located immediately to the right of an edge \tilde{e}_2 removable in ccw orientation with respect to l , $int(\tilde{f})$ is located to the right of \tilde{e}_1 , and \tilde{e}_1 and \tilde{e}_2 share a vertex.

If none of the above two situations occurs during the plane sweep, the algorithm reports that f is removable with respect to l .

The proof of correctness of this algorithm consists of two parts. The first part, f is not removable with respect to l if the algorithm finds one of the two above mentioned situations is easy to see. It remains to prove that f is removable with respect to l if none of the above mentioned situations occur. We prove this by contradiction and assume that f is not removable with respect to l although the two situations did not occur. Hence, $\exists p \in CW(f)$, such that $(cwarcl', p, cw\alpha(p)) \setminus \{p\} \cap CCW(f) \neq \emptyset$. Since the transformation from f to \tilde{f} maintains the distance from l' , $\exists p \in CW(\tilde{f})$, such that $(cwarcl', p, cw\alpha(p)) \setminus \{p\} \cap CCW(\tilde{f}) \neq \emptyset$. Let the point $q = (cwarcl', p, cw\alpha(p)) \setminus \{p\} \cap CCW(\tilde{f})$. Since every edge of \tilde{f} is strictly monotone and has a monotone derivative, p and q are located on distinct edges \tilde{e}_1 and \tilde{e}_2 of \tilde{f} respectively. Two situations are possible. First, $cw\alpha(p)$ is the cw angle that p is rotated by prior to leaving the interior of \tilde{f} . Assume without loss of generality that there is no edge between \tilde{e}_1 and \tilde{e}_2 . This assumption is valid, because if there is

an edge between \tilde{e}_1 and \tilde{e}_2 , we can find an edge \tilde{e}_2' with the desired property. Hence, \tilde{e}_1 is located immediately to the right of \tilde{e}_2 and $\text{int}(\tilde{f})$ is located to the left of \tilde{e}_1 . This corresponds to the first situation mentioned above and therefore contradicts the assumption. Second, $\text{cw}\alpha(p)$ is an arbitrarily small angle, i.e. $\text{int}(\tilde{f})$ is located to the right of the edge \tilde{e}_1 and the distance between \tilde{e}_1 and \tilde{e}_2 is arbitrarily small. Hence, \tilde{e}_1 and \tilde{e}_2 share a vertex. This corresponds to the second situation mentioned above and therefore contradicts the assumption and proves the correctness of the algorithm.

For a face with s vertices, labeling all of the edges as removable in cw or ccw orientation with respect to l and transforming f to \tilde{f} takes time $O(s)$. The plane sweep algorithm takes time $O(s \log s)$ [21, Chapter 2]. Since no two adjacent faces of P are parallel, every vertex of P is part of at most one face perpendicular to l . Hence, it takes $O(n \log n)$ total time to test all of P 's faces with inner normal \vec{n} , $l \cap S(f) = \emptyset$, and \vec{l} is parallel to \vec{n} . \square

It remains to prove Claim 3.2.10:

Proof. The proof of this statement consists of two parts. First, we prove that f is not removable with respect to l if $\exists p \in CW(f)$, such that $(\text{cvarc}(l', p, \text{cw}\alpha(p)) \setminus \{p\}) \cap CCW(f) \neq \emptyset$. From the definition of $\text{cw}\alpha(p)$, we know that $(\text{cvarc}(l', p, \text{cw}\alpha(p)) \setminus \{p\}) \cap \partial f$ is a single point q . The assumption ensures that $q \in CCW(f)$. If $(\text{cvarc}(l', p, \text{cw}\alpha(p)) \setminus (\{p\} \cup \{q\}))$ is completely contained in the exterior of P , $\text{cw}\alpha(p)$ is an arbitrarily small angle. But this implies that p cannot be rotated by an arbitrarily small angle in cw orientation with respect to l without hitting a point of $CCW(f)$. The definition of castability implies that f is not removable. Otherwise, $(\text{cvarc}(l', p, \text{cw}\alpha(p)) \setminus (\{p\} \cup \{q\})) \in \text{int}(f)$. Since p and q need to be removed in opposite orientations with respect to l , we find at least one point $p^* \in \text{cvarc}(l', p, \text{cw}\alpha(p))$, where the valid orientation with respect to l changes from cw to ccw when traversing $\text{cvarc}(l', p, \text{cw}\alpha(p))$ starting at p . As the traversal from p to q is a rotation in cw orientation around l' , p^* penetrates F_2 (respectively F_1) when rotated infinitesimally in cw (respectively ccw) orientation with respect to l . Therefore, f is not removable with respect to l .

The second part we prove is that there exists a valid partitioning of f if

$$(cwar\alpha(l', p, cw\alpha(p)) \setminus \{p\}) \cap CCW(f) = \emptyset \quad \forall p \in CW(f).$$

We prove this explicitly by assigning a unique valid orientation with respect to l to every point $q \in f$, such that no cw point of f penetrates F_2 when rotated by an arbitrarily small angle in cw orientation with respect to l and no ccw point of f penetrates F_1 when rotated by an arbitrarily small angle in ccw orientation with respect to l . Consider an arbitrary point $p \in CW(f)$. Since $(cwar\alpha(l', p, cw\alpha(p)) \setminus \{p\}) \cap CCW(f) = \emptyset \quad \forall p \in CW(f)$, every point q on $(cwar\alpha(l', p, cw\alpha(p)))$ can rotate by an arbitrarily small angle in cw orientation with respect to l without hitting a point of $CCW(f)$. Hence, we label every point $q \in int(f)$ with $\exists p \in CW(f)$, such that $q \in cwar\alpha(l', p, cw\alpha(p))$ as removable in cw orientation with respect to l . Every point $q \in int(P)$ with $q \notin cwar\alpha(l', p, cw\alpha(p)) \quad \forall p \in CW(f)$ can leave $int(f)$ by a ccw rotation with respect to l without intersecting $CW(f)$. Every point $q \in \partial P$ with $q \notin cwar\alpha(l', p, cw\alpha(p)) \quad \forall p \in CW(f)$ can rotate by an arbitrarily small angle in ccw rotation with respect to l without intersecting $CW(f)$. Hence, we label every point $q \in f$ with $q \notin cwar\alpha(l', p, cw\alpha(p)) \quad \forall p \in CW(f)$ as removable in ccw orientation with respect to l . Since the two sets $F_1 = \{q \mid \exists p \in CW(f), \text{ such that } q \in cwar\alpha(l', p, cw\alpha(p))\}$ and $F_2 = \{q \mid q \notin cwar\alpha(l', p, cw\alpha(p)) \quad \forall p \in CW(f)\}$ have the properties that $F_1 \cap F_2 = \emptyset$ and $f = F_1 \cup F_2$, a valid partitioning of f was found. Note that the set of faces forming F_1 and F_2 , respectively, consists of faces bounded by edges of f and circular arcs with center point l' . \square

This yields the conclusion that all the faces of a polyhedron P with combinatorial complexity n can be labeled as removable in cw, ccw, or no orientation with respect to a line of rotation l in space in $O(n \log n)$ total time.

Testing Connectivity

As in Section 3.2.1, it remains to test whether all the faces of P that are removable in cw orientation with respect to l and all the faces of P that are removable in ccw orientation with respect to l form connected sets, respectively. In the preceding

section, an algorithm is presented to partition and label all of P 's faces. Hence, Lemma 3.2.4 still holds for the general casting problem.

Testing Collisions

The results presented in the two previous sections allow to test whether a polyhedron P can be partitioned into two connected sets ∂P_1 and ∂P_2 , such that all faces of ∂P_1 are removable in cw orientation with respect to l and all faces of ∂P_2 are removable in ccw orientation with respect to l . As in Section 3.2.1, it remains to test whether ∂P_1 can be rotated by an arbitrarily small angle in cw orientation with respect to l without colliding with ∂P_2 and whether ∂P_2 can be rotated by an arbitrarily small angle in ccw orientation with respect to l without colliding with ∂P_1 . Lemma 3.2.5 presented in Section 3.2.1 presents an algorithm that performs this test in $O(n)$ time.

This yields the following result:

Theorem 3.2.11. *Given a polyhedron P with combinatorial complexity n and a line of rotation l in 3-dimensional space, it is possible to determine whether P is castable with respect to l in time $O(n \log n)$.*

3.3 Determining all valid casting lines

In this section, we solve the problem of finding all lines l in 3-dimensional space, such that a given polyhedron is (robustly) castable with respect to l . Two aspects of the same problem are considered: reporting a representative for each class of combinatorially distinct (robust) valid casting lines for a given polyhedron and preprocessing P such that, for any given query line l , we can efficiently determine if P is (robustly) castable with respect to l . Two lines l_1 and l_2 are combinatorially distinct if the valid casting line l_1 with respect to P cannot continuously be moved to l_2 without changing the type of removability (see Lemma 3.2.2) of any face of P .

Section 3.3.1 discusses the problem of determining all robust valid casting lines for a polyhedron. Section 3.3.2 discusses the problem of determining all valid casting lines for a polyhedron.

3.3.1 Robust rotational casting

In this section, we solve the problem of finding all lines l in 3-dimensional space, such that a given polyhedron is robustly castable with respect to l .

Recall that a polyhedron P is not robustly castable with respect to a line l with the property that some face or reflex edge of P is contained in a plane perpendicular to l . Hence, this section only considers lines l , such that no face or reflex edge of P is perpendicular to l .

The 3-dimensional casting problem can be reduced to the 2-dimensional version discussed in Chapter 2. Consider an arbitrary plane π orthogonal to l and compute the set $P' = \pi \cap P$. The intersection consists of sets of polygons, as P is a polyhedron. Hence, the castability of P' with respect to the point $l' = \pi \cap l \in \mathbb{R}^2$ can be analyzed using the methods of Chapter 2. As P' consists of a set of polygons, P' is castable with respect to l' if and only if all of the polygons in P' are castable with respect to l' . If a set of polygons forms a polygon with holes, P is not robustly castable with respect to l , because the polygon forming the hole cannot be removed without penetrating $\text{int}(P)$.

As P is a polyhedron, the boundary ∂P is an orientable surface. The proof of Lemma 3.2.4 states that P is not castable if ∂P is not connected. Hence, we assume in the following that ∂P is a connected orientable surface in \mathbb{R}^3 .

Definition 3.3.1. *The genus g of a connected orientable surface $S \in \mathbb{R}^3$ is the maximum number of cuttings along closed simple curves without disconnecting S .*

Definition 3.3.2. *A handle is a subset of a manifold that is topologically equivalent to a ball.*

The genus of a connected orientable surface S equals the number of handles of S . A *hole* in S exists if there exists a continuous loop L that cannot be homotopically deformed onto a point within the surface S . For each handle of S , there exists a hole in S , such that the corresponding loop L is partially located on the handle.

Lemma 3.3.3. *A polyhedron $P \in \mathbb{R}^3$ is robustly castable with respect to a line of rotation l if and only if every cross section P' of P with a plane π perpendicular to l is castable with respect to the point $l' = \pi \cap l$.*

Proof. The proof consists of two parts. First, P is not robustly castable with respect to l if any cross section P' is not castable with respect to l' because the rotation of ∂P around l includes the rotation of every possible $\partial P'$ around l' .

Proving that P is robustly castable with respect to l if every cross section P' is castable with respect to l' requires showing that the near and far points of $\partial P'$ form connected chains that partition ∂P into exactly two connected sets. It follows then that all the faces of one component are removable in cw orientation with respect to l and all the faces of the other component are removable in ccw orientation with respect to l . Since the partitioning of ∂P is at near and far points that are not reflex vertices, Lemma 3.2.5 ensures that no collisions occur.

To examine the location of near and far points of $\partial P'$, the plane π perpendicular to l is swept over P in direction \vec{l} . When π is swept over P , l' does not move and P' changes continuously. Hence, the near and far points of $\partial P'$ with respect to l' move continuously along edges of ∂P and orthogonal projections of l on faces of ∂P . This implies that vertices of ∂P and the start and end points of the orthogonal projections of l on faces of ∂P are the only event points of the sweep algorithm. The output of the plane sweep is a set of circular lists representing cycles on ∂P formed by near and far points of $\partial P'$ with respect to l' . The circular lists contain event points of the sweep algorithm ordered by their appearance on the cycle. During the plane sweep, a set of lists L that represent parts of circular lists is updated. When referring to event points of the plane sweep, only event points that were not swept yet are considered.

First, assume that only one event point occurs per cross section. The following events can occur at an event point v of the plane sweep:

1. In P' , a new polygon occurs as isolated vertex v . Hence, v is both a near and a far point with respect to l' . A new list that represents part of a circular list is created and added to the set L . It contains the event point adjacent to v with minimal distance from l (a near point with respect to l'), v , and the event point adjacent to v with maximal distance from l (a far point with respect to l') in that order.
2. In P' , an existing polygon disappears as isolated vertex v . Hence, v is both a

near and a far point with respect to l' . The event point v is already contained in one list of L as first point and in one (possibly the same) list of L as last point. If v is contained in two different lists, the lists are joined. Otherwise, the list of L containing v as both first and last point is stored as circular list.

3. In P' , an existing polygon splits into two polygons. The two polygons have the common vertex v , and therefore, v is both a near and a far point with respect to l' . A new list that represents part of a circular list is created and added to the set L . It contains the event point adjacent to v with minimal distance from l , v , and the event point adjacent to v with maximal distance from l in that order.
4. In P' , two polygons merge into one. The two polygons have the common vertex v , and therefore, v is both a near and a far point with respect to l' . The event point v is already contained in one list of L as first point and in one (possibly the same) list of L as last point. If v is contained in two different lists, the lists are joined. Otherwise, the list of L containing v as both first and last point is stored as circular list.
5. No topological changes occur in the cross section and the event point v is not contained in any list of L . Hence, v is neither a near nor a far point with respect to l' and no updates are required.
6. No topological changes occur in the cross section and the event point v is contained in a list of L as first or last point respectively. If v is the first point, v is a near point with respect to l' . The event point adjacent to v with minimal distance from l is added to the list of L containing v at the front. If v is the last point, v is a far point with respect to l' . The event point adjacent to v with maximal distance from l is added to the list of L containing v at the tail.

Note that although twelve topological changes can occur when a plane is swept over a polyhedron (see [11]), only the first four above are relevant as polygons with holes are not castable.

If a cross section contains more than one event point, the event points can simply be treated in sequential order, because no two event points are on one face or edge of P . As the set of circular lists L returned by the plane sweep represents chains of near and far points of $\partial P'$ with respect to l' , ∂P is partitioned along L . Hence, it remains to show that L partitions ∂P into exactly two connected sets. We prove this by induction on the genus g of ∂P .

Base case: There are no split or merge vertices. A simple polyhedron P (i.e. $g = 0$) is robustly castable with respect to l if every cross section P' is castable with respect to l' . The plane sweep returns one simply connected cycle. We prove this by induction on the number of split and merge event points (event points of type 3 and 4).

- Base case: The first event point is of type 1 and creates one list that represents part of a circular list. As long as only event points of type 5 and 6 occur, no new lists are created. As there are no merge event points, it is impossible that further event points of type 1 occur. Hence, the only topological change that will occur when the last event point is swept is of type 2. This results in one circular list representing a simply connected cycle on ∂P . As ∂P has genus 0, any simply connected cycle partitions ∂P into exactly two connected sets.
- Induction step: Given that the plane sweep algorithm returns one circular list if k topological changes of type 3 or 4 occur during the plane sweep, we show that the plane sweep algorithm returns one circular list if $k + 1$ topological changes of type 3 or 4 occur during the plane sweep. If the $k + 1^{st}$ topological change is a split, a new list L_k is created. After the split, the two new polygons in P' disappear in event points of type 2. Note that they cannot merge again as P is simply connected. When the first of the two new polygons in P' disappears in an event point of type 2, L_k joins another list of L . The initial assumption implies that this list is eventually the only remaining list that will be returned as circular list. If the $k + 1^{st}$ topological change is a merge, a new polygon occurs as an isolated vertex at an event point of type 1 creating a new list L_k . When this new polygon merges with another polygon, L_k joins another list of

L . The initial assumption implies that this list is eventually the only remaining list that will be returned as circular list. Hence, the sweep algorithm always returns one simply connected circular list for a polyhedron of genus 0.

Induction step: Given that a polyhedron P_g with genus g is robustly castable with respect to l if every cross section P'_g is castable with respect to l' , we show that a polyhedron P_{g+1} with genus $g + 1$ is robustly castable with respect to l if every cross section P'_{g+1} is castable with respect to l' . Adding a handle to the castable polyhedron P_g yields P_{g+1} . Hence, only event points incident to the hole adjacent to the newly created handle are relevant. The first event point that occurs is of type 3, i.e. a new list L_g is created. As long as only event points of type 5 and 6 occur, no new lists are created. It is impossible that an event point of type 1 or 2 occurs incident to a hole. If multiple event points of type 3 occur, the polygons in P'_{g+1} can therefore only merge again. Hence, one simply connected circular list representing a cycle on ∂P_{g+1} is returned when the sweep plane passed the hole. As all of the event points are incident to the hole adjacent to the newly created handle, the cycle does not intersect any of the existing cycles. Hence, P_{g+1} is partitioned into exactly two connected sets by the points in L , and is therefore robustly castable with respect to l . It remains to prove that by adding a handle to P_g , every possible polyhedron P_{g+1} of genus $g + 1$ can be created. The Principal Theorem of surface topology [36, §39] states that two closed orientable surfaces are topologically equivalent if and only if they have the same genus. Hence, any polyhedron of genus $g + 1$ is topologically equivalent to P_{g+1} and can therefore be created by adding a handle to P_g . \square

In analogy to the 2-dimensional casting problem discussed in Chapter 2, it is possible to define black regions. For a polyhedron, black regions exist for faces, reflex edges, and reflex vertices.

Definition 3.3.4. *For a face f of P , the black region of f is defined as $S^+(f)$.*

Note that there is no valid casting line for f that properly intersects the black region of f (see Lemma 3.2.2). Figure 3.2 shows the black region of f .

Definition 3.3.5. Let e be a reflex edge of P and denote the inner normals of its two adjacent faces by \vec{n}_1 and \vec{n}_2 . Define the near wedge of e as $S^+(e) = \{\vec{p} + t_1\vec{n}_1 + t_2\vec{n}_2 : p \in cl(e) \text{ and } t_1, t_2 \geq 0\}$ and the far wedge of e as $S^-(e) = \{\vec{p} + t_1\vec{n}_1 + t_2\vec{n}_2 : p \in cl(e) \text{ and } t_1, t_2 < 0\}$. The black region of a reflex edge e is defined as $(S^+(e) \cup S^-(e)) \setminus e$.

Definition 3.3.6. A vertex v of P is a reflex vertex of P if and only if all of v 's adjacent edges are reflex edges of P . Let v be a reflex vertex of P and denote the inner normals of its r adjacent faces by $\vec{n}_1, \dots, \vec{n}_r$. Define the near cone of v as $S^+(v) = \{\vec{v} + \sum_{i=1}^r t_i\vec{n}_i : t_i > 0 \forall i = 1, \dots, r\}$ and the far cone of v as $S^-(v) = \{\vec{v} + \sum_{i=1}^r t_i\vec{n}_i : t_i < 0 \forall i = 1, \dots, r\}$. The black region of a reflex vertex v is defined as $S^+(v) \cup S^-(v)$.

Figure 3.3 illustrates the black region of a reflex edge and of a reflex vertex.

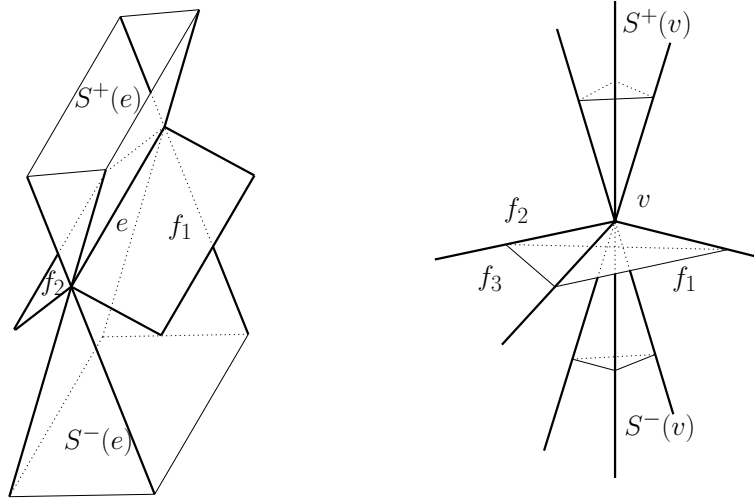


Figure 3.3: Left: The near and far wedge of e . Right: The near and far cone of v .

The black regions of faces, reflex edges, and reflex vertices of P are the *black regions induced by P* .

Lemma 3.3.7. A polyhedron P is robustly castable with respect to a line of rotation l , if and only if l does not intersect any of the black regions induced by P .

Proof. The proof consists of two parts. First, we prove that P is not robustly castable with respect to l if l intersects any of the black regions induced by P . If l intersects the black region induced by a face f of P , $l \cap S^+(f) \neq \emptyset$ and Lemma 3.2.2 states that P is not castable with respect to l . If l intersects the black region of a reflex edge e of P , there exists a point q on e that locally minimizes or maximizes the Euclidean distance from ∂P to l . If l is parallel to e , this is true for every point on e . Consider the intersection of P with the plane π perpendicular to l passing through q . In the cross-section, q is a reflex vertex that has extremal distance to $\pi \cap l$. Hence, $\pi \cap l$ is contained in the two-dimensional black region of q . Lemmata 2.3.14 and 3.3.3 ensure that P is not robustly castable with respect to l . If l intersects the black region induced by a reflex vertex v of P , v locally minimizes or maximizes the Euclidean distance from ∂P to l . With an argument similar to the one we used for reflex edges, we can then show that P is not robustly castable with respect to l .

Second, we prove that P is robustly castable with respect to l if l does not intersect the union of black regions induced by P . We prove this by contradiction and assume that l does not intersect the union of black regions induced by P and that P is not robustly castable with respect to l . Since P is not robustly castable with respect to l , Lemma 3.3.3 ensures that there exists a cross section of P with a plane π perpendicular to l that is not castable with respect to $\pi \cap l$. Hence, there exists a cross section, such that $\pi \cap l$ is contained in the 2-dimensional black region of the cross section. Three cases are possible:

1. The point $l' = \pi \cap l$ is contained in the black region of an edge $f' = \pi \cap f$, where f is a face of P with normal \vec{n} not parallel to the direction \vec{l} of l . Denote the perpendicular projection of l' on $\text{int}(f')$ by q . Since the normal vector of f' is the projection of \vec{n} onto the plane π , there exists a $t \geq 0$, such that

$$\vec{l}' = \vec{q} + \left(\vec{n} - \frac{\vec{n} \cdot \vec{l}}{\vec{l} \cdot \vec{l}} \vec{l} \right) t.$$

Consider the point l^* with

$$\vec{l}^* = \vec{l}' + t \frac{\vec{n} \cdot \vec{l}}{\vec{l} \cdot \vec{l}} \vec{l} = \vec{q} + t \vec{n}.$$

Clearly, $l^* \in l$, as l' reaches l^* by being moved in direction \vec{l} only. Since $q \in f$, l^* is also contained in the black region of f . This contradicts the initial assumption.

2. The point $l' = \pi \cap l$ is contained in the black region of a reflex vertex $e' = \pi \cap e$, where e is a reflex edge of P . Since $\pi \cap e$ is a vertex, none of the two faces adjacent to e is perpendicular to l . Denote the normal vectors of e 's two adjacent faces by \vec{n}_1 and \vec{n}_2 respectively. There exist two constants t_1, t_2 with $t_1, t_2 \geq 0, t_1 t_2 \neq 0$ or $t_1, t_2 < 0$, such that

$$\vec{l}' = \vec{e}' + \left(\vec{n}_1 - \frac{\vec{n}_1 \cdot \vec{l}}{\vec{l} \cdot \vec{l}} \vec{l} \right) t_1 + \left(\vec{n}_2 - \frac{\vec{n}_2 \cdot \vec{l}}{\vec{l} \cdot \vec{l}} \vec{l} \right) t_2.$$

Consider the point l^* with

$$\vec{l}^* = \vec{l}' + t_1 \frac{\vec{n}_1 \cdot \vec{l}}{\vec{l} \cdot \vec{l}} \vec{l} + t_2 \frac{\vec{n}_2 \cdot \vec{l}}{\vec{l} \cdot \vec{l}} \vec{l} = \vec{e}' + t_1 \vec{n}_1 + t_2 \vec{n}_2.$$

Clearly, $l^* \in l$, as l' reaches l^* by being moved in direction \vec{l} only. Since $e' \in e$, l^* is also contained in the black region of e . This contradicts the initial assumption.

3. The point $l' = \pi \cap l$ is contained in the black region of a reflex vertex $v' = \pi \cap v$, where v is a reflex vertex of P . Since v' is a reflex vertex, π intersects exactly two of v 's adjacent faces and neither of those faces is perpendicular to l . Hence, a proof similar to the previous proof shows that l intersects the black region of v . This contradicts the initial assumption.

Since every possible case yields a contradiction, the initial statement is proven to be true. \square

Consider all the planes bounding black regions of faces, reflex edges, and reflex vertices of P . The black region of a face f of P is bounded by $deg(f)$ planes and can be expressed using $deg(f)$ lines perpendicular to f passing through vertices of f . The black region of a reflex edge e of P can be expressed using five lines: four lines perpendicular to the adjacent faces of e passing through the vertices of e and the supporting line of e . Finally, the black region of a reflex vertex v of P can be expressed using $deg(v)$ lines perpendicular to the faces adjacent to v passing through v . Hence,

the arrangement A in \mathbb{R}^3 containing for every face f of P the lines perpendicular to f passing through vertices of f and the supporting lines of the reflex edges of P describes all of P 's black regions. The arrangement A contains $O(n)$ lines, since $\sum_{f \in P} \deg(f) = 2E$, where E is the number of edges of P . Two distinct lines $g_1, g_2 \in A$ are in the same *equivalence class* of A if and only if it is possible to move g_1 to g_2 without crossing any of the lines forming A . If g_1 does not cross any of the lines forming A during the transformation, it cannot enter or leave any of the black regions induced by P . Assume that g_1 and g_2 are in the same equivalence class of A . Lemma 3.3.7 ensures that g_2 is a valid casting line for P if and only if g_1 is a valid casting line for P .

Reporting all valid casting lines

In this section, we find and report a representative for each class of combinatorially distinct valid robust casting lines for a given polyhedron P with combinatorial complexity n in time $O(n^4\alpha(n))$, where $\alpha(n)$ is the inverse Ackermann function. The main idea used in this section is the fact that a line l in space is a valid robust casting line for P if and only if l does not intersect any black region induced by P , see Lemma 3.3.7.

Theorem 3.3.8. *Given a polyhedron P with combinatorial complexity n , it is possible to report all of the valid robust casting lines for P in \mathbb{R}^3 in time $O(n^4\alpha(n))$, where $\alpha(n)$ is the inverse Ackermann function.*

Proof. To report all of the valid casting lines for P in \mathbb{R}^3 , we construct the arrangement A of the lines defined as intersections of the planes bounding black regions induced by P . Every cell of A corresponds to exactly one equivalence class of A . Therefore, it is possible to label each cell c of A as an equivalence class of valid or invalid casting lines for P . Finally, a representing line for each equivalence class labeled as valid is reported.

The arrangement A of $O(n)$ lines is constructed using a method by McKenna and O'Rourke [30, 31]. They represent k lines in \mathbb{R}^3 using four parameters per line and construct $O(k^2)$ planar arrangements of hyperbolas in $O(k^2\alpha(k))$ time each. They

show that the arrangement has complexity $\Theta(k^4)$ and can be constructed in time $O(k^4\alpha(k))$. The arrangement is represented as a graph G , where every line touching four of the k given lines or parallel to one of the k given lines and touching two of the remaining $k - 1$ given lines is represented as a node of G . Using this method, A has complexity $O(n^4)$ and can be constructed in time $O(n^4\alpha(n))$.

Once A and G have been constructed, every face of G needs to be labeled as valid or invalid. For this purpose, a boolean value is associated with every face f , reflex edge e , and reflex vertex v of P that indicates whether the current equivalence class of lines intersects the black region of f , e , or v respectively. We start at an arbitrary face f_G of G and test for each face, reflex edge, and reflex vertex of P whether it causes the equivalence class of lines represented by f_G to be invalid. After testing, we set the boolean value of each face, reflex edge, and reflex vertex appropriately and compute the number b of faces, reflex edges, and reflex vertices that cause f_G to be invalid. Clearly, f_G is valid if and only if $b = 0$. This computation takes $O(n)$ time as every face, reflex edge, and reflex vertex of P needs to be considered. Next, G is traversed in depth-first order. Each time, an edge e_G of G is crossed, we update the boolean values of the face, the reflex edge, and the reflex vertex of P that induce e_G and the counter b . This way, every face of G is labeled in constant time a piece. The edge e_G and its incident nodes represent valid casting lines of P if and only if one or more of e_G 's adjacent faces is labeled valid. Hence, all of the equivalence classes of A can be labeled in time $O(n^4)$ and it is possible to report a representative for each class of combinatorially distinct valid casting lines for P in time $O(n^4\alpha(n))$. \square

After computing A and G in $O(n^4\alpha(n))$ preprocessing time, it is possible to perform line location in A , i.e. answer the question whether a given line l is a valid robust casting line for P . However, this query takes $O(n)$ time [31]. Note that it is possible to report whether P is robustly castable with respect to a line l in time $O(n)$ without preprocessing using Theorem 3.2.6. Therefore, a query time of $O(\log n)$ is preferable. Hence, we use another approach to preprocess space for fast line location.

Preprocessing space for fast line location

In this section, we preprocess the polyhedron P in $O(n^{4+\epsilon})$ expected time in a way that allows to answer whether a query line is a valid casting line for P in time $O(\log n)$, where ϵ is an arbitrarily small positive constant.

Using the same main idea as above, consider all the planes bounding black regions of faces, reflex edges, and reflex vertices of P and their $O(n)$ intersecting lines. The arrangement of the lines is now represented in a way that allows fast queries.

For this purpose, each line is represented using Plücker coordinates, proposed in 1868 by Plücker [33]. In Plücker space, an oriented 3-dimensional line is described as a point called Plücker point in oriented projective 5-dimensional space using six coordinates. Dually, each line can also be represented as a hyperplane called Plücker hyperplane in oriented projective 5-dimensional space. For two lines l and g , the Plücker point of l is located on the Plücker hyperplane of g if and only if l and g intersect and vice versa. That way, intersection tests of lines are linearized at the cost of operating in projective 5-dimensional space.

Consider a line l passing through the two points $p = (p_x, p_y, p_z, p_w)$ and $q = (q_x, q_y, q_z, q_w)$ given in homogeneous space. The Plücker coordinates of the Plücker point corresponding to l are $[l_0, l_1, l_2, l_3, l_4, l_5]$ and the Plücker coefficients of the Plücker hyperplane corresponding to l are $[l_5, -l_4, l_3, l_2, -l_1, l_0]$ with $l_0 = p_x q_y - p_y q_x$, $l_1 = p_x q_z - p_z q_x$, $l_2 = p_y q_z - p_z q_y$, $l_3 = p_x q_w - p_w q_x$, $l_4 = p_y q_w - p_w q_y$, and $l_5 = p_z q_w - p_w q_z$.

Note that the vector $\vec{l}_n = [l_0, -l_1, l_2]^T$ is the cross product $\vec{p} \times \vec{q}$, which is the normal vector of the plane passing through the line l and the origin. Furthermore, the vector $\vec{l}_d = [l_3, l_4, l_5]^T$ is the difference vector $\vec{p} - \vec{q}$, which is the direction of the oriented line l .

The coordinates $[l_0, l_1, l_2, l_3, l_4, l_5]$ specify a line in Plücker space if and only if

$$l_0 l_5 - l_1 l_4 + l_2 l_3 = 0. \quad (3.1)$$

This implies that not every set of 6-dimensional coordinates corresponds to a line in 3-dimensional space. In fact, in Plücker space, the set of all the lines corresponds to a 4-dimensional hypersurface.

Two lines l and g intersect if and only if

$$l_0g_5 - l_1g_4 + l_2g_3 + l_3g_2 - l_4g_1 + l_5g_0 = 0. \quad (3.2)$$

If the Plücker point of l is not located on the Plücker hyperplane of g , the sign of Equation (3.2) indicates the orientation of l in relation to g .

A survey on Plücker coordinates can be found in Stolfi's book [38, Chapter 19]. Chazelle et al. [14, 15] applied Plücker coordinates in Computational Geometry to represent arrangements of lines in space. However, as they are only interested in small parts of the full arrangement, such as the envelope of the arrangement or single cells of the arrangement, they do not construct the full arrangement of lines in space.

Theorem 3.3.9. *A polyhedron P with combinatorial complexity n can be preprocessed in $O(n^{4+\epsilon})$ expected time into a data structure of size $O(n^{4+\epsilon})$ such that for any given line l , we can decide in $O(\log n)$ time if P is robustly castable with respect to l , where ϵ is an arbitrarily small positive constant.*

Proof. In a first step, all of the faces of P are triangulated. Since each face f of P is a possibly non-simple polygon, finding the constrained Delaunay triangulation and ignoring edges in the exterior of f yields a triangulation of f . Chew [16] and Wang and Schubert [40] independently proposed methods to find the constrained Delaunay triangulation of k vertices in time $O(k \log k)$. Since each edge has exactly two adjacent faces, it requires total time $O(n \log n)$ to triangulate all of P 's faces. Since the triangulation adds $O(k)$ new edges to each face, the combinatorial complexity of the triangulated polyhedron is $O(n)$.

The black regions of the triangular faces of P are defined in analogy to the black regions of faces of P before the triangulation. Let parallel lines intersect at infinity and augment the boundary of the black regions induced by P by vertices at infinity in every direction. The black region of a face of P is now bounded by four triangles. Furthermore, the black region of a reflex edge of P is bounded by eight triangles and the black region of a reflex vertex of P is bounded by s triangles, where s equals twice the degree of the vertex.

Observe that any line intersecting the black region induced by P must intersect the boundary of the black region induced by P , since none of the black regions induced by

faces, reflex edges, and reflex vertices of P are unbounded along a line. As explained above, the boundary of the black region induced by P can be represented by $O(n)$ triangles.

The question of whether a given query line intersects the black region induced by P is equivalent to the question of whether the query line intersects any of the bounding triangles of the black regions induced by P . Note that this problem is similar to the ray shooting problem arising in computer graphics, where a query ray is shot into a scene containing a set of triangles and the aim is to report the first triangle of the set hit by the query ray. However, our problem is easier than ray shooting, because we are only interested in knowing whether the query line intersects any of the triangles bounding the black regions induced by P .

For each of the $O(n)$ edges of bounding triangles of the black region induced by P , the supporting lines are represented as hyperplanes in Plücker space. We then follow the approach proposed by de Berg et al. [20] to answer whether a given query line intersects any of the bounding triangles of the black region induced by P . Aronov et al. [3] showed that the complexity of the intersection of the arrangement of n Plücker hyperplanes in projective 5-dimensional space with the Plücker quadric expressed in Equation (3.1) is $O(n^4 \log n)$. Since we know that every query line is represented by a point inside the Plücker quadric, it suffices to consider the arrangement of hyperplanes representing supporting lines of triangles in the zone of the Plücker quadric. De Berg et al. use the randomized approach by Clarkson [17] to construct this 4-dimensional arrangement. Denote by H the set of hyperplanes representing supporting lines of triangles bounding the black regions induced by P . Let $R \subset H$ be a sample of size $O(r)$ with the property that any cell of the triangulated arrangement in projective 5-dimensional space is intersected by at most $\frac{n}{r} \log r$ hyperplanes of R . Clarkson proved that such a sample can be obtained in $O(nr^5)$ expected time by sampling repeatedly until the condition is met. A tree is built recursively as follows. The root of the tree is the arrangement $A(R)$ induced by R in the zone of the Plücker quadric. Each of the $O(r^4 \log r)$ children of the root corresponds to a cell of $A(R)$. If for a child of the root, there exists a triangle that is intersected by any line represented by a Plücker point in that cell, the child is labeled as invalid region and the recursion stops. Otherwise,

the cell is recursively split and labeled. De Berg et al. [20] show that, by choosing r appropriately, the expected time and total space requirement to build this tree is $O(n^{4+\epsilon})$, where ϵ is an arbitrarily small positive constant.

A given query line is represented as a point in Plücker space and point location is performed in the tree. When a leaf is reached, the label of the leaf reports whether the line intersects any of the triangles bounding the black region induced by P . Since a search in the tree requires $O(\log n)$ time and the label can be read in constant time, the query time of the algorithm is $O(\log n)$. \square

3.3.2 General rotational casting

In this section, we solve the problem of finding all lines l in 3-dimensional space, such that a given polyhedron P is castable with respect to l . Since the situation is identical to the situation in Section 3.3.1 if lines perpendicular to a face or an edge of P are neglected, this section focuses on lines perpendicular to a face or an edge of P .

As in Section 3.3.1, consider a plane π perpendicular to the line of rotation l and intersect π with P . The intersection $P' = \pi \cap P$ consists of proper intersections, edges of P and faces of P . If a cross section P' does not contain any faces perpendicular to P , P' can still be examined as before. Recall that if P' contains a face of P , we need to test whether that face is removable with respect to l' . The test can be performed using Lemma 3.2.9. Hence, Lemma 3.3.3 can be changed to the following:

Lemma 3.3.10. *A polyhedron P in \mathbb{R}^3 is castable with respect to a line of rotation l if and only if every cross section P' of P with a plane π perpendicular to l consists of castable and removable polygons with respect to $l' = \pi \cap l$.*

Proof. The main part of this Lemma was proven in the proof of Lemma 3.3.3. Hence, only faces perpendicular to l are discussed here. The proof of Lemma 3.2.9 shows that P is not castable if any face perpendicular to P is not removable with respect to l . Hence, P is not castable with respect to l if any of the cross sections P' contains a polygon that is not castable or not removable with respect to l' .

Next, we show that P is castable with respect to l if every cross section P' consists of castable and removable polygons with respect to l' . The same approach as in the

proof of Lemma 3.3.3 is used and only faces perpendicular to l are discussed. If a cross section contains a face f perpendicular to l , we know that there exists a valid partitioning of f . If a new polygon occurs in P' as face f , treat the face like a vertex of type 1. If an existing polygon disappears in P' as face f , treat the face like a vertex of type 2. If an existing polygon splits into two or more polygons in P' , f contains holes. Every arc contributing to a valid partitioning of f (see Lemma 3.2.9) that connects two holes of f is treated as a vertex of type 3. Arcs connecting to the outer face of f are treated as vertices of type 6. If two or more polygons merge into one in P' , f contains holes. Every arc contributing to a valid partitioning of f that connects two holes of f is treated as a vertex of type 4. Arcs connecting to the outer face of f are treated as vertices of type 6. In each of the cases, the set of edges and arcs yielding a valid partitioning of f can be treated in sequential order. Hence, every case can be represented as a vertex of type 1 to 6. Therefore, the proof of Lemma 3.3.3 holds even when the line of rotation is perpendicular to a face or reflex edge of P . \square

After introducing lines perpendicular to a face or edge of P , Lemma 3.3.7 does not hold any more. There are two reasons for this. First, there exist situations where P is castable with respect to l although l intersects black regions induced by P . As this cannot happen when faces and edges orthogonal to l are neglected (see Lemma 3.3.7), only those two situations need to be considered. If l intersects the black region of a face f perpendicular to l , f is not removable with respect to l (see Lemma 3.2.2). Hence, P is not castable with respect to l . If l intersects the near wedge of a reflex edge e perpendicular to l , consider the cross-section of P with the plane π perpendicular to l containing e . The point $l' = l \cap \pi$ is contained in the 2D black region of e . Hence, Lemma 3.3.3 implies that P is not castable with respect to l . However, if l only intersects the far wedge of a reflex edge e of P perpendicular to l , P is castable with respect to l , because e can be split on the perpendicular projection of l on e and one part of e is removable in cw orientation with respect to l and the other part of e is removable in ccw orientation with respect to l . Second, there exist lines of rotation l , such that l does not intersect any of the black regions induced by P and P is not castable with respect to l . This can only occur, if P contains a face

perpendicular to l that is not removable with respect to l (see Lemma 3.3.7). An example where a face perpendicular to l prevents P from being castable although l does not intersect any of the black regions induced by P is shown in Figure 3.4.

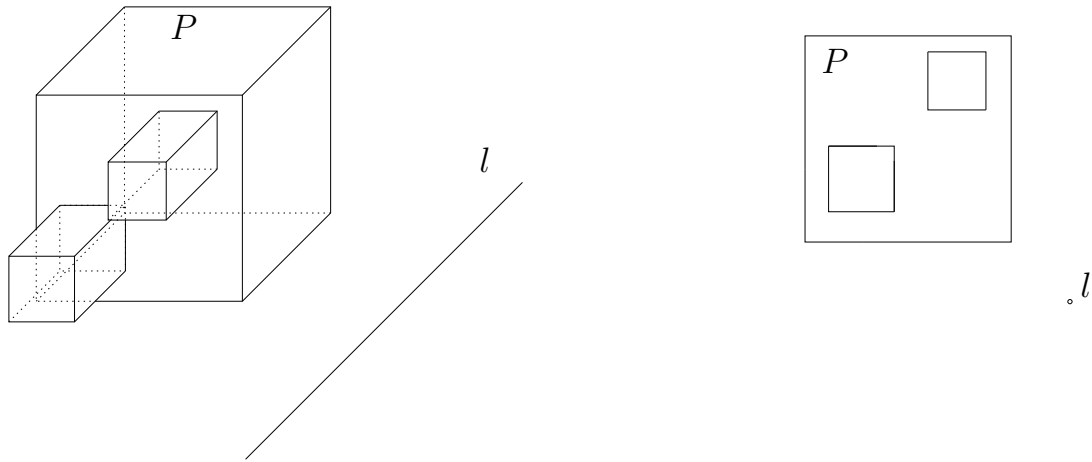


Figure 3.4: Example of polyhedron P that is not castable with respect to l although l does not intersect the black regions induced by P . Figure shows perspective view and front view.

Hence, Lemma 3.3.7 can be restated the following way:

Lemma 3.3.11. *Let P be a polyhedron and let l be a line in space. The polyhedron P is castable with respect to l if l does not intersect any of the black regions induced by P and if every face of P perpendicular to l is removable with respect to l . If l intersects the black region induced by any face of P , the black region induced by any reflex vertex of P , or the near wedge of any reflex edge of P , P is not castable with respect to l .*

In analogy to Section 3.3.1, the aim is now to construct an arrangement A with the property that every equivalence class of A represents exactly one class of combinatorially distinct lines. To achieve this, the arrangement A used in Section 3.3.1 needs to be augmented to handle lines perpendicular to reflex edges and faces of P .

Reporting all valid casting lines

In this section, we modify the results of Section 3.3.1 to find and report a representative for each class of combinatorially distinct valid casting lines for a given polyhedron P in time $O(n^4 \log n)$.

Theorem 3.3.12. *Given a polyhedron P with combinatorial complexity n , it is possible to report all of the valid casting lines for P in \mathbb{R}^3 in time $O(n^4 \log n)$.*

Proof. In Section 3.3.1, the arrangement A represented by the graph G introduced by McKenna and O'Rourke [31] is constructed to represent all the classes of combinatorially distinct lines in space. The lines that are not perpendicular to a reflex edge or face of P can still be categorized and labeled using this technique. However, the arrangement needs to be extended to handle lines perpendicular to reflex edges and faces of P .

We will first discuss how to augment the arrangement to represent lines perpendicular to reflex edges of P . Recall that for any line perpendicular to a reflex edge of P , the far wedge of the reflex edge does not induce a black region. Hence, we need to insert all of the lines in space that are perpendicular to an edge of P into the arrangement A . The aim is to insert nodes into G , such that any time a line g that is not perpendicular to an edge e of P is moved continuously until g is perpendicular to e , a node of G is passed. McKenna and O'Rourke [31] prove that a line restricted to intersect three skew lines has one degree of freedom. This degree of freedom can be used to move the line until it is perpendicular to e , unless every line touching the three skew lines is perpendicular to e . Consider a line touching three of the lines in the arrangement A . If every line touching the three skew lines is perpendicular to e , no node is inserted to G . The reason is that G already contains nodes that ensure that any time a line g that is not perpendicular to e is moved continuously until g is perpendicular to e , a node of G is passed. If there exists a line touching the three skew lines that is not perpendicular to e , the line touching the three skew lines perpendicular to e is inserted as node in G . For three given skew lines and an edge e , this line can be found in constant time, since every line touching three skew lines has one degree of freedom. To insert all of these nodes in G , the edges of G are

traversed in depth-first order. An edge of G describes lines that are touching three skew lines. For each edge of G , find the $O(n)$ nodes describing a line touching three skew lines and perpendicular to an edge e of P . This requires $O(n)$ time. Then, sort the nodes according to their appearance on the edge in $O(n \log n)$ time and insert them in order. It takes $O(n \log n)$ time to insert the new nodes for one edge of G . Since G has $O(n^3)$ edges, the total time to construct this arrangement is $O(n^4 \log n)$.

This arrangement needs to be augmented further to handle lines perpendicular to faces of P correctly. Lemma 3.3.11 implies that for any line l perpendicular to a face f of P , P is only castable if f is removable with respect to l . For each face f of P , the arrangement A contains at least three lines perpendicular to f that bound the black region of f . Furthermore, the arrangement A contains every line touching four given lines or touching three given lines, where the third line is touched at infinity. Hence, G already contains all the nodes that represent lines perpendicular to faces of P .

It remains to label the faces of the new graph G we constructed. To label every face of G , traverse G in depth-first order. We store all the counters explained in Section 3.3.1 and an additional counter c of the number of far wedges of reflex edges perpendicular to lines in f_G that intersect the lines in f_G . This counter can be updated in constant time, if we store three boolean values with each reflex edge e : one value indicating whether the current line is in the near wedge of e , one value indicating whether the current line is in the far wedge of e , and one value indicating whether the current line is perpendicular to e . Furthermore, we store a boolean variable indicating whether the lines in f_G are perpendicular to a face f of P . If the lines represented by f_G are not perpendicular to any face of P , simply proceed as in Section 3.3.1. Initialize the counters once in $O(n)$ time and update them in constant time per face. After subtracting c from the total number of black regions for f_G , we label f_G as valid if and only if the total number of black regions equals zero. If the lines represented by f_G are perpendicular to a face f of P , it is required to test whether f is removable with respect to the lines represented by f_G . For this purpose, we need to construct a planar arrangement for each face of P . Then, if the total number of black regions computed before equals zero, we need to look in the planar arrangement whether f

is removable with respect to l .

For each set of faces F of P with $f \in F$ has inner normal $\pm\vec{n}$, we construct a planar arrangement that subdivides a plane π with inner normal \vec{n} into points $l' = l \cap \pi$ that are intersections of lines l in direction \vec{n} , such that each $f \in F$ is removable with respect to l and points $l' = l \cap \pi$ that are intersections of lines l in direction \vec{n} , such that $\exists f \in F$ that is not removable with respect to l . Hence, every point l' in the plane π represents the line l perpendicular to π that intersects π in l' . To subdivide π into valid and invalid regions, several steps are required. First, the $O(n)$ lines bounding black regions of P are projected to π . The planar arrangement has complexity $O(n^2)$. Every face of this arrangement can be labeled as black or non-black region, respectively, in $O(n^2)$ time by walking through the arrangement and maintaining a counter of all intersected black regions using a similar technique as explained in Theorem 3.3.8. For every face of the arrangement labeled non-black, we need to test for each $f \in F$ whether f is removable with respect to lines represented by that face. Let p_1 and p_2 denote two arbitrary points inside the same face labeled as non-black of the arrangement. Since p_1 and p_2 are in the same face of the arrangement, all the faces of P have the same valid orientation with respect to the two lines represented by p_1 and p_2 , that is, the faces are either removable in cw orientation with respect to both lines, removable in ccw orientation with respect to both lines, or need to be split with respect to both lines. Hence, if the algorithm described in Lemma 3.2.9 is used to determine whether f is removable with respect to the line represented by p_1 and p_2 , respectively, all the edges of f are labeled in the same way in the first step. Hence, the two transformed polar planes with origins p_1 and p_2 , respectively, have the same structure. Although the actual distances to p_1 and p_2 are different, the same sequence of monotone curves with monotone derivatives occurs in the transformed plane. Furthermore, if an edge is split, the actual split vertex is different, but the situation in the transformed plane remains the same. Therefore, f is removable with respect to p_2 if and only if f is removable with respect to p_1 . This implies that for each face of the arrangement labeled as non-black, we need to test one representative using the algorithm of Lemma 3.2.9 in $O(r \log r)$ time, where r is the number of edges of f . Let s denote the cardinality of F . In $O(rs \log r)$ time, it

is tested whether each $f \in F$ is removable with respect to lines represented by one face of the arrangement labeled as non-black. Hence, it requires $O(n^2rs \log r)$ time to construct and label the arrangement for the faces F . This step needs to be done for each of the $O(n)$ sets of faces with codirectional inner normals. In the worst case, both s and r are in the order of n . This seems to imply that the time complexity to construct and label the arrangements for all faces of P is $O(n^5 \log n)$. However, each face of P is represented in exactly one planar arrangement and each edge of P is shared by exactly two faces. Therefore, the sum of all the cardinalities and the sum of all the edges are both in the order of n . This implies that the total time complexity to construct and label the arrangements for all faces of P is $O(n^4 \log n)$.

This implies that in $O(n^4 \log n)$ time, we can find and report all of the combinatorially distinct valid casting lines for P . \square

After computing A and G in $O(n^4 \log n)$ preprocessing time, it is possible to perform line location in A , i.e. answer the question whether a given line l is a valid casting line for P . A query in A requires $O(n)$ time [31]. If the face of A contains the information that l is perpendicular to a set F of faces of P , it is necessary to query the arrangement induced by F to determine whether each $f \in F$ is removable with respect to l . It is possible to determine $l' = \pi \cap l$ in constant time and to determine the face of the arrangement induced by F that contains l' in $O(\log n)$ time [29]. Hence, it takes $O(n + \log n) = O(n)$ time to report whether P is castable with respect to l . Note that this implies that we can report faster whether P is castable with respect to l than without preprocessing (see Theorem 3.2.11). However, a query time of $O(\log n)$ is preferable. Hence, we use another approach to preprocess space for fast line location.

Preprocessing space for fast line location

In this section, we preprocess P in expected time $O(n^{4+\epsilon})$ in a way that allows to answer whether a query line is a valid casting line for P in time $O(\log n)$, where ϵ is an arbitrarily small positive constant.

As before, consider all of the planes bounding the black regions induced by P

and their $O(n)$ intersecting lines. Again, we use ray shooting in Plücker space to determine whether the boundary of any black region is intersected by the query line. However, the data structures become more complex, since lines perpendicular to faces and reflex edges of P need to be taken into consideration.

Theorem 3.3.13. *A polyhedron P with combinatorial complexity n can be preprocessed in $O(n^{4+\epsilon})$ expected time into a data structure of size $O(n^{4+\epsilon})$ such that, for any given line l , we can decide in $O(\log n)$ time if P is castable with respect to l , where ϵ is an arbitrarily small positive constant.*

Proof. We employ three data structures to answer whether a query line l is a valid casting line for P . The first data structure represents the black regions induced by faces of P , the black regions induced by reflex vertices of P , and the far wedges induced by reflex edges of P . By Lemma 3.3.11, any query line intersecting the boundary of the black regions represented in the first data structure is an invalid casting line for P . Hence, we can use the same approach as in the proof of Lemma 3.3.9 to build a data structure of size $O(n^{4+\epsilon})$ in expected time $O(n^{4+\epsilon})$ that answers whether l intersects any of the boundaries of black regions represented in the first data structure in $O(\log n)$ time. If the answer to the query reports that l intersects any of the black regions represented by the first data structure, l is an invalid casting line and we are done. Otherwise, the second data structure is queried.

The second data structure considers the far wedges of the reflex edges of P . For all of the far wedges of reflex edges of P that are not perpendicular to l , de Berg et al.'s data structure [20] can be used as in the proof of Lemma 3.3.9 to answer whether l intersects the boundary of those far wedges. Recall that de Berg et al.'s data structure has size $O(n^{4+\epsilon})$ and answers queries in $O(\log n)$ time.

Each reflex edge of P is associated with four triangles bounding the far wedge of that reflex edge. The direction of a reflex edge of P can be represented as a point on the upper unit hemisphere of directions, because the orientation of the edge can be neglected. Using central projection from the upper unit hemisphere onto the plane $\pi : z = 1$ with the origin as center of projection yields a representation of the directions of the reflex edges of P in 2-dimensional Euclidean space. Note that each

reflex edge is represented as a point in π , which may lead to more than one reflex edge mapping to one point in π . This representation has the property that directions that lie on a great arc of the upper unit hemisphere of directions correspond to points lying on a line in π . The directions in the upper unit hemisphere perpendicular to l can be represented as a great arc of the upper unit hemisphere of directions or as a line l' in π . Hence, all of the reflex edges of P that are not perpendicular to l can be found using two half-space queries in π with the two open half-spaces bounded by l' . The triangles bounding the far wedges of the reflex edges of P that are not perpendicular to l can be used in de Berg et al.'s data structure [20] to answer whether l intersects the boundary of those far wedges.

For the second data structure, note that we have a set of $O(n)$ points in 2-dimensional Euclidean space associated with four triangles each. We can therefore use a two-level data structure based on a cutting tree [21, Chapter 16]. A $\frac{1}{r}$ -cutting in the plane is a collection of pairwise disjoint planar simplices that cover the entire plane with the property that each simplex intersects at most $\frac{n}{r}$ of the n given lines. A cutting tree on n lines in the plane is recursively defined as follows: if $n = 1$, then the tree consists of a single node storing the line. Otherwise, a $\frac{1}{r}$ -cutting is found and for each simplex of the $\frac{1}{r}$ -cutting, we store a child of the current node. That child contains the simplex that created the child, a list of all of the lines located completely above the simplex, and a list of all of the lines located completely below the simplex. Furthermore, each child is recursively defined as the root of a cutting tree on the lines crossing the simplex associated with the child [21, Chapter 16]. Chazelle [13] proved that a $\frac{1}{r}$ -cutting in the plane of size $O(r^2)$ can be found in $O(nr)$ time.

The cutting tree used in the two-level data structure stores the set H of lines dual to the 2-dimensional points representing the reflex edges of P . Every node of the cutting tree stores a set $H' \subset H$ of lines corresponding to a set of triangles bounding the far wedges of the reflex edges associated with the lines H' . These triangles are stored in de Berg et al.'s data structure [20]. Van Kreveld's results [39, Theorem 5.8] imply that, for an appropriate choice of r , the total size of the cutting tree storing de Berg et al.'s data structure in each node is $O(n^\epsilon n^{4+\epsilon})$ for any $\epsilon > 0$. Replacing ϵ by $\frac{\epsilon}{2}$ yields the space requirement $O(n^{\frac{\epsilon}{2}} n^{4+\frac{\epsilon}{2}}) = O(n^{4+\epsilon})$.

To analyze the time requirement to build the second data structure, we use that a $\frac{1}{r}$ -cutting in the plane of size $O(r^2)$ can be found in $O(nr)$ time [13]. We use the fact that by a similar analysis as the one used by van Kreveld [39, Theorem 5.8], this yields the following recurrence for the expected construction time $P(n)$ of the data structure: $P(n) \leq c_1 r^2 P(\frac{n}{r}) + c_2 r^{c_3} g(n) + c_4 nr$, where c_1, \dots, c_4 are constants and where $g(n) = n^{4+\epsilon}$ is the expected preprocessing time required to build de Berg et al.'s data structure [20]. By choosing r to be a sufficiently large constant, this recurrence solves to $O(n^\epsilon(n^2 + g(n)) + n^2 \log n)$ for any $\epsilon > 0$. If we replace ϵ by $\frac{\epsilon}{2}$, the data structure therefore requires $O(n^{4+\epsilon})$ expected construction time.

For a query line l , we perform two queries in that data structure. The line l' in π that represents all the directions perpendicular to l is found. We query the data structure with both open half-spaces bounded by l' . This is done by querying the data structure twice with the dual point of l' and by reporting all of the nodes of the cutting tree representing only lines of H that lie on the correct side of the dual query point, respectively. Then, l is represented as Plücker point and a query is performed in each of the secondary structures corresponding to the reported nodes. Van Kreveld's results [39, Theorem 5.8] imply that this data structure supports queries in $O(\log n)$ time.

If the answer to the query reports that l intersects any of the black regions represented by the second data structure, l is an invalid casting line and we are done. Otherwise, the third data structure is queried.

The third data structure tests whether each face of P perpendicular to the query line l is removable with respect to l . As in the proof of Theorem 3.3.12, for each set of faces F of P with $f \in F$ has inner normal $\pm\vec{n}$, we construct a planar arrangement that subdivides a plane with inner normal \vec{n} into points $l' = l \cap \pi$ that are intersections of lines l in direction \vec{n} , such that each $f \in F$ is removable with respect to l and points $l' = l \cap \pi$ that are intersections of lines l in direction \vec{n} , such that $\exists f \in F$ that is not removable with respect to l . We know from the proof of Theorem 3.3.12 that in $O(n^4 \log n)$ time, we can build the planar arrangements with total space requirement $O(n^3)$ for all the sets F . For each set F , represent the directions $\pm\vec{n}$ in a dictionary with the property that the set F defined by a given direction \vec{d} can found in $O(\log n)$

time, if it exists.

Querying this data structure is slightly more complicated than in the proof of Theorem 3.3.12. For a query line l , query the dictionary with the direction of the query line to find the set of faces F of P perpendicular to l . If there are no faces perpendicular to l , l is a valid casting line for P . Otherwise, compute the intersection l' of l with the planar arrangement associated with F and perform point location. This requires $O(\log n)$ time [29]. Retrieving the label of the face of the planar arrangement containing l' answers whether all of the faces of F are removable with respect to l . Hence, querying the third data structure takes time $O(\log n)$.

Hence, the total space requirement and expected preprocessing time for the three data structures is $O(n^{4+\epsilon})$. Queries can be answered in $O(\log n)$ total time. \square

Chapter 4

Conclusion and Future Work

The worst thing you can do to a problem is solve it completely.

DANIEL KLEITMAN

We have studied the problem of clamshell casting in both two and three dimensions, where the boundary of the object to manufacture is piecewise linear. In two dimensions, an algorithm was developed to solve the problem of determining whether a polygon with n vertices is castable with respect to a given point in the plane with running time $O(n)$. Furthermore, two algorithms were developed to report all the valid centers of rotation for a given polygon in the plane. The running times of the algorithms are $O(n)$ for convex polygons and $O(n^2)$ for simple polygons in general and shown to be worst-case optimal.

In three dimensions, an algorithm was developed to solve the problem of determining whether a polyhedron of arbitrary genus with combinatorial complexity n is castable with respect to a given line in space with running time $O(n \log n)$. If the lines are restricted not to be perpendicular to a reflex edge or a face of the polyhedron, the algorithm's running time becomes $O(n)$. Furthermore, it is possible to report all of the valid casting lines for a given polyhedron in space in time $O(n^4 \log n)$. If the lines are restricted not to be perpendicular to a reflex edge or a face of the polyhedron, the algorithm's running time becomes $O(n^4 \alpha(n))$. Alternatively, the polyhedron can be preprocessed in $O(n^{4+\epsilon})$ expected time into a data structure of size $O(n^{4+\epsilon})$, such that for any given line l , we can decide in $O(\log n)$ time if l is a valid casting line.

The following interesting related problems require further research.

- The control of the physical casting machinery is imperfect. This yields to surface defects if the cast of the object slides along the boundary of the object. The algorithms presented in this thesis should therefore be extended to determine whether an object is castable in a way that the cast does not slide along the boundary of the object.
- Since many objects do not have a piecewise linear boundary, the algorithms should be extended to handle more general object boundaries.
- The definition of clamshell casting used in this thesis only tests whether the cast of an object with piecewise linear boundary can be opened by an arbitrarily small angle without breaking the object or the cast. To physically manufacture the object, it is required that the cast can be opened by a sufficiently large angle to remove the object from the cast without breaking the object or the cast. This problem is difficult, since the object can be removed from the cast by an arbitrary sequence of transformations.
- In this thesis, the boundary of the object is defined to be the cast. In case of rotations around arbitrarily small angles, this model is sufficient. However, when considering larger angles of rotations, the thickness of the cast has an influence on the maximum angle of rotation that does not break the object or the cast. Hence, the cast needs to be assigned a thickness.

Bibliography

- [1] H. Ahn, O. Cheong, and R. van Oostrum. Casting a polyedron with directional uncertainty. *Computational Geometry*, 26:129–141, 2003.
- [2] H. Ahn, M. de Berg, P. Bose, S. Cheng, D. Halperin, J. Matousek, and O. Schwarzkopf. Separating an object from its cast. *Computer-Aided Design*, 34:547–559, 2002.
- [3] B. Aronov, M. Pellegrini, and M. Sharir. On the zone of a surface in a hyperplane arrangement. *Discrete and Computational Geometry*, 9:177–186, 1993.
- [4] H. Bartsch. *Taschenbuch Mathematischer Formeln, 18. Auflage*. Fachbuchverlag Leipzig, 1999.
- [5] P. Bose. Geometric and computational aspects of manufacturing processes. *Ph.D. thesis, McGill University*, 1994.
- [6] P. Bose, D. Bremner, and M. van Kreveld. Castability of simple polyhedra. *Algorithmica: Special Issue on Manufacturing*, 19:84–113, 1997.
- [7] P. Bose, P. Morin, M. Smid, and S. Wuhrer. Rotational clamshell casting in three dimensions. Technical Report TR0604, Carleton University, 2006.
- [8] P. Bose, P. Morin, M. Smid, and S. Wuhrer. Rotational clamshell casting in two dimensions. Technical Report TR0603, Carleton University, 2006.
- [9] P. Bose and G. Toussaint. Geometric and computational aspects of manufacturing processes. *Computers and Graphics (invited paper)*, 18(4):487–497, 1994.

- [10] P. Bose and G. Toussaint. Geometric and computational aspects of gravity casting. *Computer Aided Design*, 27:455–464, 1995.
- [11] P. Bose and M. van Kreveld. Generalizing monotonicity: On recognizing special classes of polygons and polyhedra by computing nice sweeps. *International Journal of Computational Geometry*, 15(6):591–608, 2005.
- [12] P. Bose, M. van Kreveld, and G. Toussaint. Filling polyhedral molds. *Computer-Aided Design: Special Issue on Manufacturing*, 30(4):245–254, 1998.
- [13] B. Chazelle. Cutting hyperplanes for divide-and-conquer. *Discrete and Computational Geometry*, 9:145–158, 1993.
- [14] B. Chazelle, H. Edelsbrunner, L. Guibas, and M. Sharir. Lines in space-combinators, algorithms and applications. In *STOC '89: Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 382–393. ACM Press, 1989.
- [15] B. Chazelle, H. Edelsbrunner, L. Guibas, M. Sharir, and J. Stolfi. Lines in space : Combinatorics and algorithms. *Algorithmica*, 15:428–447, 1996.
- [16] P. Chew. Constrained delaunay triangulations. *Algorithmica*, 4:97–108, 1989.
- [17] K. Clarkson. New applications of random sampling in computational geometry. *Discrete and Computational Geometry*, 2:195–222, 1987.
- [18] T. Corman, C. Leiserson, R. Rivest, and C. Stein. *Introduction to Algorithms Second Edition*. The MIT Press, 2001.
- [19] R. Creese. *Introduction to Manufacturing Processes and Materials*. Publisher: Marcel Dekker, 1999.
- [20] M. de Berg, D. Halperin, M. Overmars, J. Snoeyink, and M. van Kreveld. Efficient ray shooting and hidden surface removal. In *7th Symposium on Computational Geometry*, pages 51–60, 1991.

- [21] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry Algorithms and Applications Second Edition*. Springer-Verlag, 1991.
- [22] S. Fekete and J. Mitchell. Geometric aspects of injection molding. In *Workshop Geometric Comput. Aspects Injection Molding*, 1993.
- [23] U. Finke and K. Hinrichs. Overlaying simply connected planar subdivisions in linear time. In *SCG '95: Proceedings of the eleventh annual symposium on Computational geometry*, pages 119–126, New York, NY, USA, 1995. ACM Press.
- [24] J. Goodman and J. O'Rourke. *Handbook of Discrete and Computational Geometry, Second Edition*. Chapman & Hall CRC, 2004.
- [25] K. Hoffmann, K. Mehdorn, P. Rosenstiehl, and R. Tarjan. Sorting jordan sequences in linear time using level-linked search trees. *Information and Control*, 68:170–184, 1986.
- [26] K. Hui and S. Tan. Mold design with sweep operations - a heuristic search approach. *Computer Aided Design*, 24:81–91, 1992.
- [27] M. Keil. A simple algorithm for determining the envelope of a set of lines. *Information Processing Letters*, 39(3):121–124, 1991.
- [28] H. Keisler. *Elementary Calculus An Infinitesimal Approach Second Edition*. Prindle, Weber, and Schmidt, 1986.
- [29] D. Kirkpatrick. Optimal search in planar subdivisions. *SIAM Journal on Computing*, 12(1):28–35, 1983.
- [30] M. McKenna. *Applications of Arrangements to Geometric Problems in Higher Dimensions*. Ph.D Dissertation, John Hopkins University Baltimore, Maryland, 1989.
- [31] M. McKenna and J. O'Rourke. Arrangements of lines in 3-space: A data structure with applications. In *Proceedings of the fourth annual symposium on Computational geometry*, pages 371 – 380, New York, NY, USA, 1988. ACM Press.

- [32] J. O'Rourke. *Computational Geometry in C, Second Edition*. Cambridge University Press, New York, 1998.
- [33] J. Plücker. *Neue Geometrie des Raumes*. B.G. Teubner, Leipzig, 1868.
- [34] F. Preparata and K. Supowit. Testing a simple polygon for monotonicity. *Information Processing Letters*, 12(4):161–164, 1981.
- [35] A. Rosenbloom and D. Rappaport. Moldable and castable polygons. *Computational Geometry Theory and Applications*, 4:219–233, 1994.
- [36] H. Seifert and W. Threlfall. *A textbook of topology*. Academic Press, 1980.
- [37] B. Simpson. *History of the Metal-Casting Industry, Second Edition*. American Foundrymen's Association, 1969.
- [38] J. Stolfi. *Oriented Projective Geometry - A Framework for Geometric Computations*. Academic Press Professional, Inc., 1991.
- [39] M. van Kreveld. *New results on data structures in computational geometry*. Ph.D Dissertation, Utrecht University, 1992.
- [40] C. Wang and L. Schubert. An optimal algorithm for constructing the Delaunay triangulation of a set of line segments. In *Proceeding of the third Annual ACM Symposium on Computational geometry*, pages 223–232, 1987.