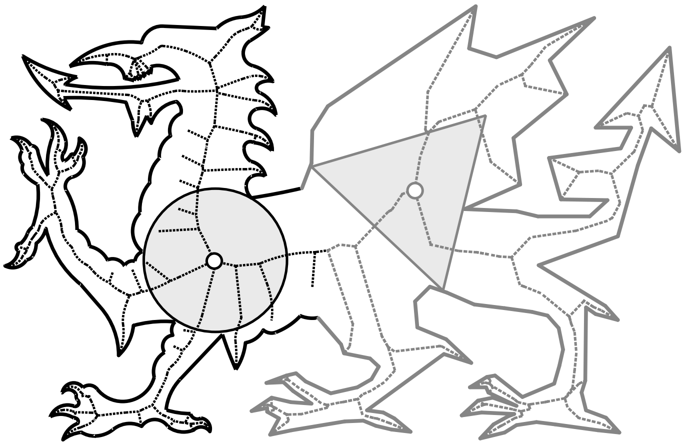


The medial axis of geometric objects for various convex metrics

Doctoral Thesis
at
Graz University of Technology

submitted by

Wolfgang Aigner

August 2, 2011

Institute for Software Technology

Faculty of Computer Science,
Graz University of Technology
A-8010 Graz, Austria

Advisor: Dipl.-Ing. Dr.techn. Associate Prof. Oswin Aichholzer

Acknowledgements

I want to thank my family in general which always provided me a safe haven when I felt the need to flee the big city of Graz, and my parents in particular, who made my academic education possible in the first place by supporting me (not only) financially.

No less I want to thank my dear Karin, who admittedly quit reading through this thesis after the abstract (as if her logic and verification mambo jambo was any more fun...), but still has been my most important moral support during the last years.

From the academic point of view I probably owe the most part of my progress to (what I call them off the record) the triumvirate of Computational Geometry. Thus I sincerely thank

Oswin Aichholzer for being my supervisor and adviser, who I could rely on when it came to any difficulties whatsoever and who redefined for me the notion of “networking”,

Franz Aurenhammer for luring me into the realm of CG, who was always up for a good chat and a good beer and who writes papers as thrilling to read as a spy novel,

and last but not least Bert Jüttler who supported me financially when the roof was on fire, and creatively when things seemed so often hopeless.

For their valuable contributions I would also like to express my gratitude to my other co-authors, namely (in no particular order) Thomas Hackl, Nicola Wolpert, Günter Rote, Margot Rabl, Elisabeth Pilgerstorfer and Katerina Čech Dobiášová.

Finally I owe many thanks to my colleagues and friends in Graz and Linz, who were always available for questions, discussions and the occasional after-work get-together, which is in my opinion the icing on the cake :)

Abstract

A geometric shape in the plane or in space is usually considered to be an open connected set $\Omega \subset \mathbb{R}^d$ for $d = 2, 3$, and there exist various ways to define and represent such an object.

A possibility for describing such a shape is by its boundary $\partial\Omega$. This boundary can be given in an exact way as a collection of algebraic curves and surfaces (e.g. piece-wise linear). If no exact representation is possible, then approximations of $\partial\Omega$ can be considered, realized by e.g. point representations. Shapes with piece-wise linear boundaries are called polygons in the plane, and polyhedra in space. Especially in 2D, properties and structures (e.g. skeletal structures) of such objects are well-known and efficient algorithms for computing them do exist. But already by replacing the straight line components of such a polygon by circular arcs we enter almost uncharted territory in many contexts. In 3D, even for simple boundary representations the complexity of many applications (e.g. offset computation) is usually rather high, also concerning algebraic issues.

For many of these problems a different representation of the shape is helpful: the medial axis transform. The medial axis $\mathrm{MA}(\Omega)$ of a shape Ω consists of the union of all centers of maximal inscribed disks in the plane, and maximal inscribed spherical balls in space. The medial axis transform $\mathrm{MAT}(\Omega)$ associates with every point of $\mathrm{MA}(\Omega)$ the radius of the corresponding maximal disk (ball), and thus fully represents Ω.

On one hand we want to be able to compute the medial axis for a wide set of shapes, on the other hand we want to reduce its complexity especially concerning the algebraic degree of its primitives. Thus the two main tasks of this thesis are:

1. The set of shapes for medial axis computation in two dimensions are extended by allowing circular boundary representations. For that matter we describe a divide-and-conquer approach which allows efficient medial axis computation, and in the progress even offers the possibility for Voronoi computation with respect to complex sites.

2. The difficulty of exact medial axis computation for triangulated solids in 3D is reduced by introducing a piece-wise linear metric. The provided algorithm computes the medial axis (which is also piece-wise linear) by its projections on the boundary contacts of the shape.

The two approaches provide an exact medial axis of the respective class of shapes and allow convenient trimmed offset computation. In both cases, the medial axis computation as well as the offset computation are fully implemented.

Keywords: medial axis, Voronoi diagram, piece-wise linear, exact computation, offset computation, divide and conquer, bisector computation, convex distance functions

Zusammenfassung

Eine geometrische Form in der Ebene oder im Raum ist eine offene, zusammenhängende Punktmenge $\Omega \subset \mathbb{R}^d$ für $d = 2, 3$, und es gibt verschiedene Möglichkeiten ein solches Objekt darzustellen.

Eine Möglichkeit um so eine Form zu beschreiben, ist ihr Rand $\partial\Omega$. Dieser Rand kann exakt als eine Menge von algebraischen Kurven und Oberflächen gegeben sein (z.B. stückweise linear). Wenn keine exakte Darstellung möglich ist, dann können Annäherungen von $\partial\Omega$ betrachtet werden die z.B. durch Punktwolken realisiert werden können. Formen mit stückweise linearem Rand werden in der Ebene Polygone genannt, und im Raum Polyeder. Speziell in 2D sind die Eigenschaften und Strukturen (z.B. das Skelett) solcher Objekte bekannt und in der Regel existieren effiziente Algorithmen für deren Berechnung. Aber schon wenn man die linearen Segmente durch Kreisbögen ersetzt betritt man in vielerlei Hinsicht Neuland. In 3D ist die Komplexität vieler Anwendungen (z.B Offsetberechnung) schon fuer sehr einfache Randdarstellungen recht hoch, auch in algebraischer Hinsicht.

Für viele dieser Probleme ist eine alternative Darstellung einer geometrischen Form sehr hilfreich: die Mittelachsen-Transformation. Die Mittelachse MA(Ω) einer Form Ω besteht aus den Mittelpunkten aller maximal-eingeschriebenen Kreisscheiben in der Ebene, und aller maximal-eingeschriebenen Bällen im Raum. Die Mittelachsen-Transformation MAT(Ω) assoziiert mit jedem Punkt von MA(Ω) den Radius der entsprechenden maximalen Scheibe (bzw. des maximalen Balles), und stellt Ω daher vollständig dar.

Auf der einen Seite ist es unser Ziel die Mittelachse für eine möglichst große Menge an Formen berechnen zu können, auf der anderen Seite wollen wir auch die Komplexität der Berechnungen möglichst gering halten. Daher sind die zwei Hauptpunkte dieser Arbeit die folgenden:

1. Die Menge der geometrischen Formen in der Ebene, für welche die exakte Mittelachse berechnet werden kann, wird durch Objekte deren Ränder aus Kreisbögen bestehen erweitert. Der entwickelte Divide-und-Conquer Algorithmus ist effizient, und kann auch für die Berechnung von komplexen Voronoi-diagrammen erweitert werden.

2. Die Komplexität der Mittelachsen-Berechnung für Polyeder mit trianguliertem Rand in 3D wird durch das Verwenden von stückweise linearen Metriken reduziert. Der Algorithmus berechnet die induzierte stückweis lineare Mittelachse über ihre Projektionen auf dem Rand des Objektes.

Beide Algorithmen berechnen die exakte Mittelachse für die entsprechende Objekt-Klasse. Außerdem sind beide einfach erweiterbar für Offset-Berechnung, was, zusätzlich zur eigentlichen Mittelachsenberechnung, auch implementiert wurde.

Deutsche Fassung:
Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008
Genehmigung des Senates am 1.12.2008

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Graz, am …………………………… ………………………………………………..
(Unterschrift)

Englische Fassung:

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

…………………………… ………………………………………………..
date (signature)

Contents

Acknowledgements **i**

Abstract **ii**

Zusammenfassung **iii**

Statutory declaration **iv**

Table of contents **v**

1 Introduction **1**
- 1.1 Contribution of this thesis . . . 3
  - 1.1.1 Chapter 2: Free-from shapes in 2D . . . 3
  - 1.1.2 Chapter 3: Piece-wise linear medial axis in 3D . . . 5
  - 1.1.3 Related publications . . . 6
- 1.2 General definitions . . . 7
  - Medial axis structure in two and three dimensions . . . 8

2 Circular boundary representation **10**
- 2.1 Preliminaries . . . 10
  - Biarc approximation . . . 10
- 2.2 Medial axis of simple shapes . . . 12
  - 2.2.1 Divide step . . . 13
    - Base cases . . . 15
  - 2.2.2 Termination arguments . . . 16
    - 2.2.2.1 Preventing redundant cuts . . . 16
    - 2.2.2.2 Special geometric set-ups . . . 18
    - 2.2.2.3 Local maximal curvature . . . 19
  - 2.2.3 Conquer step . . . 19
    - Branching points . . . 20
  - 2.2.4 Putting the pieces in place . . . 21
    - Expected runtime $\mathcal{O}(n \log n)$ . . . 21
  - 2.2.5 Implementation and examples . . . 22
    - 2.2.5.1 Implementation with CGAL . . . 22
    - 2.2.5.2 Examples . . . 25
- 2.3 Voronoi diagram via medial axis . . . 30
  - 2.3.1 Computation of medial axes with cycles . . . 31

2.3.1.1 Where to break the cycles . . . 31
2.3.1.2 Augmented domains . . . 32
2.3.1.3 How to break the cycles . . . 34
2.3.1.4 Computing the medial axis of $\mathcal{A}_{\mathcal{S}}$ . . . 35
2.3.2 Practical Aspects . . . 35
2.4 Trimmed offset computation . . . 38
2.4.1 The base case set-up . . . 38
2.4.2 Medial axis-induced subshapes . . . 39
Monotonic subshapes . . . 39
2.4.3 Partial offset computation . . . 40
2.5 Exact computation for circular boundaries with rational coefficients . . . 41
2.5.1 Rational circular arc boundary . . . 42
2.5.2 Exact construction of a maximal disk . . . 43
2.5.3 Exact bisector computation . . . 43
2.5.4 Partial axis confinement . . . 46
Apollonius: tritangent circles . . . 47
2.5.5 Handling degeneracies . . . 50
2.6 Conclusion of Chapter 2 . . . 51

3 Piece-wise linear metrics **52**
3.1 Preliminaries . . . 52
3.1.1 Piece-wise linear metric . . . 52
3.1.2 Contact types and classes . . . 54
3.1.3 MA$_{\tilde{B}}$ in the plane . . . 56
Pseudo-branchings and jump edges . . . 56
3.1.4 MA$_{\tilde{B}}$ in 3-space . . . 57
Pseudo-seams and jump sheets . . . 58
3.2 The medial axis of polygons . . . 60
3.2.1 Piece-wise and Euclidean medial axis . . . 60
3.2.1.1 Jumps . . . 60
3.2.1.2 Implicit pruning . . . 61
3.2.1.3 Convergence . . . 63
3.2.2 Combinatorial size . . . 63
3.2.2.1 Reflex and convex features . . . 64
3.2.2.2 An upper bound . . . 65
3.3 The medial axis of polyhedra . . . 68
3.3.1 Projections and arrangements . . . 68
3.3.1.1 Projections . . . 69
3.3.1.2 Contact arrangements . . . 69
3.3.2 Arrangement Computation . . . 72
3.3.2.1 Algorithm outline . . . 72
3.3.2.2 Constructing maximal* balls . . . 73
3.3.2.2.1 AABB Trees . . . 74
3.3.2.2.2 Computing the contact domain . . . 75
3.3.2.2.3 Contact domain for *EE* contacts . . . 75
3.3.2.2.4 Contact domain for *FV* contacts . . . 76
3.3.2.3 Finding projection lines . . . 77

3.3.2.3.1 Different opposite contact classes 78
3.3.2.3.2 Equal opposite contact classes 79
3.3.2.3.3 Putting things together 81
3.3.2.4 Arrangement computation: the algorithm 81
Vertex-dependent jump projections 86
3.3.3 Assembling the medial axis and offset 88
3.3.3.1 Assembling the medial axis from its projections 88
Connectivity 90
3.3.3.2 Assembling the offset from the medial axis 90
3.3.4 Implementation and Examples 92
3.3.4.1 CGAL implementation details 92
3.3.4.2 Runtime examples 94
3.4 Conclusion of Chapter 3 101

4 Summary and future work **103**

Glossary **107**

Bibliography **113**

Curriculum Vitae **114**

Chapter 1

Introduction

The description of geometric shapes and objects is an important topic in computational geometry, and plays a central role in numerous areas of related fields such as robotics, solid modeling and computer-aided design in general. Throughout this work a shape Ω is considered to be a connected open set in $\mathbb{R}^d$ for $d = 2, 3$.

A common way to represent a shape Ω is via its boundary $\partial\Omega$. Depending on the nature of the shape, it is either possible to give an exact boundary representation (e.g. piece-wise linear boundaries for polytopes or boundaries composed of more complex algebraic curves and surfaces), or it has to be dealt with an approximation of the shapes contour. Such an approximation may again be a piece-wise smooth structure (e.g. joint biarcs in the plane [52, 79] or meshes in space [18]), but also a description via point clouds [78] as resulting from 2D- and 3D-acquisition devices.

In 1967, Blum [17] introduced the medial axis as a possibility for shape description. The medial axis $\mathrm{MA}(\Omega)$ of a shape Ω is the locus of points inside the shape being equidistant (with respect to the Euclidean distance) from the contour of this shape. This is equivalent to the locus of center points of maximal inscribed balls in Ω (see Figure 1.1). The set of all these maximal inscribed balls is called medial axis transform or $\mathrm{MAT}(\Omega)$. A more formal definition is given in Section 1.2.

The union over all balls in the medial axis transform is a full representation of the shape, while the medial axis is directly linked to the shapes boundary.

The usefulness of the medial axis transform as a shape descriptor shows itself in applications such as offset computation [15, 23, 60] and mesh generation [57, 56]. The tree-like structure of the medial axis itself is commonly used for e.g. shape matching [31, 36] and manipulation (animation) [20, 38] or motion planning [11, 12]. Almost every application brings a different boundary representation to work with, requiring different approaches for the computation of the medial axis.

For non-exact shape representations the proximity of the obtained medial axis approximation to the axis of the original shape is usually one of the main quality criteria. For boundary-representing point samples it is a common approach to obtain a medial axis approximation by considering a subset of the Voronoi diagram [8, 35]. However, like for polygonal approximations of a shape, these approaches have to deal with a lot of unwanted artefacts which require expensive pruning [14, 24]. The scale axis approach [54], processing shapes that are represented by finite sets of balls, tries to achieve the pruning of unnecessary parts by care-

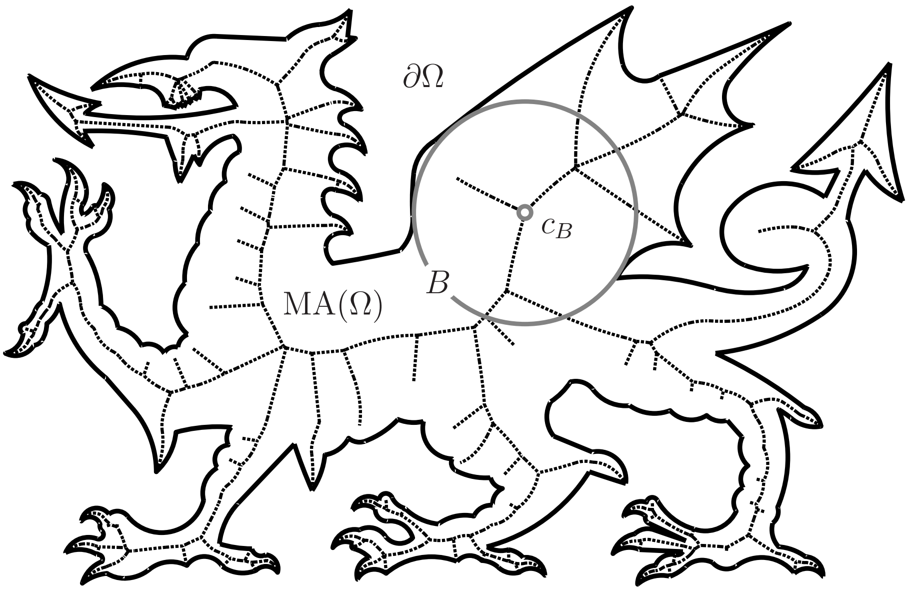


Figure 1.1: The picture shows a shape represented by its boundary $\partial\Omega$ together with its medial axis $\mathrm{MA}(\Omega)$. The center c_B of the maximal inscribed ball B lies on the medial axis.

ful scaling of these balls. This, however, can lead to the introduction of other topologically incorrect fragments if not done with care.

For medial axis computation of a given exact shape boundary there exist two principal problems that need to be addressed.

1. Determination of the combinatorial structure of the axis.
2. Computation of the bisectors.

The simpler the boundary representation, the easier these two tasks can be accomplished.

In 2D, algorithms for polygonal boundaries in [27, 61, 76, 86] are fast in general. Bisectors in this case are line segments and parabolic curves. In space, bisectors of linear primitives are quadric surfaces, the trisectors (being bisector intersections), are curves of algebraic degree already up to 4 [48]. Even for piece-wise linear boundaries, exact medial axis computation therefore requires complex algebraic algorithms [33].

Even in the plane, for curved boundaries, bisector computation is, due to numeric and algebraic complexity (see [29] for a short overview of previous work), usually a hard problem. Quite a lot of work has been devoted to this geometric aspect of the medial axis. See, for example, [42] where focus lies on rational boundary curves, and [34] where curvature properties are utilized for treating cubic boundary splines. A popular approach is local tracing [30, 58], where the medial axis is calculated by tracing either the shape boundary or the axis bisectors. In particular, so-called predictor/corrector methods [23, 34] have been proposed for approximating the medial axis in a piecewise manner.

In addition to the problematic evaluation of the bisectors often comes the nontrivial determination of the axis' combinatorial structure. For curved boundary objects in the plane, most theoretically fast algorithms compute the entire Voronoi diagram, and afterwards prune away unwanted and incorrect features. Complicated merging, or insertion, steps have to be performed, depending on whether the algorithm was based on divide-&-conquer [66, 86], or on incremental insertion [10, 50, 51]. As such steps process previously computed parts of the medial axis, they are numerically involved and subject to errors if not implemented with care [49].

Many of the above-mentioned approaches are limited to the plane. Most of the implemented algorithms in 3D are computing approximations of the medial axis for data that does not fully represent the shape, as point samples [35] or sets of balls [6, 54]. In the case of polyhedral objects, there exist numerical tracing techniques [19] (which have recently been extended to objects with curved boundaries [59]) and methods based on spatial decompositions [47, 77].

While the above methods for polyhedral input have (to my knowledge) not been used in practice and often only provide approximations of the medial axis, the algorithm described in [33] is fully implemented and computes the algebraically exact medial axis of a polyhedron. The problematic issue here is the complexity of the combinatorial structure for larger inputs as well as the high algebraic degree of the axis primitives, resulting to some extent from the Euclidean unit ball. Bisector computation by the use of more general distance functions [25, 67] or in particular polyhedral ones [26] has up to this point only been considered from a more theoretical point of view.

1.1 Contribution of this thesis

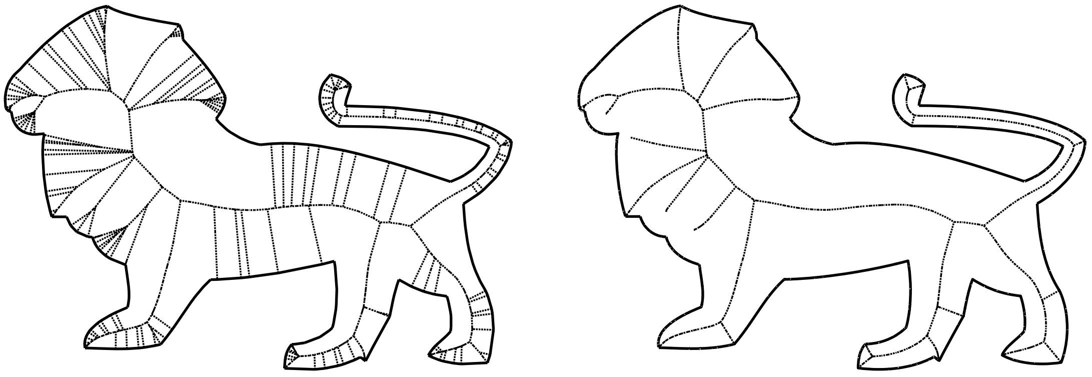
This thesis contributes two principal approaches with rather different objectives to the field of medial axis computation.

1.1.1 Chapter 2: Free-from shapes in 2D

Most of the approaches described above that deal with curved input are rather theoretical work. A recent practical one in the plane has been the extension of the VRONI Voronoi code [61] for points and line segments to circular arcs [62]. The program is fast with an expected $\mathcal{O}(n \log n)$ runtime, however the underlying algorithm is incremental insertion and endpoints of the circular arcs have to be inserted prior to their defining objects. In addition the computation of the medial axis of a smooth object with VRONI would require heavy pruning at the joined points of the bounding arcs.

In Section 2.2 we will describe a divide-and-conquer approach for medial axis computation of simple free-form shapes (represented by biarc approximations) in the plane (published as [2]). By choosing an appropriate biarc approximation scheme [52, 80] the monotonicity of curvature of the original shape as well as its basic topology can be preserved, which makes the computed medial axis converge to the exact one. See Figure 1.2a and Figure 1.2b for a demonstration of the advantages of a biarc approximation over a, e.g., polygonal one. As the medial axis of a shape with piece-wise circular boundary is composed of conic arcs it has the same analytic complexity as for polygonal domains.

The shape decomposition approach [28] is adopted to achieve simplicity and numerical robustness of the algorithm. As decomposition is by maximal inscribed balls (disks in the plane), it is naturally suited to shapes with piecewise circular boundaries. The resulting



(a) Polygon approximation

(b) Biarc approximation

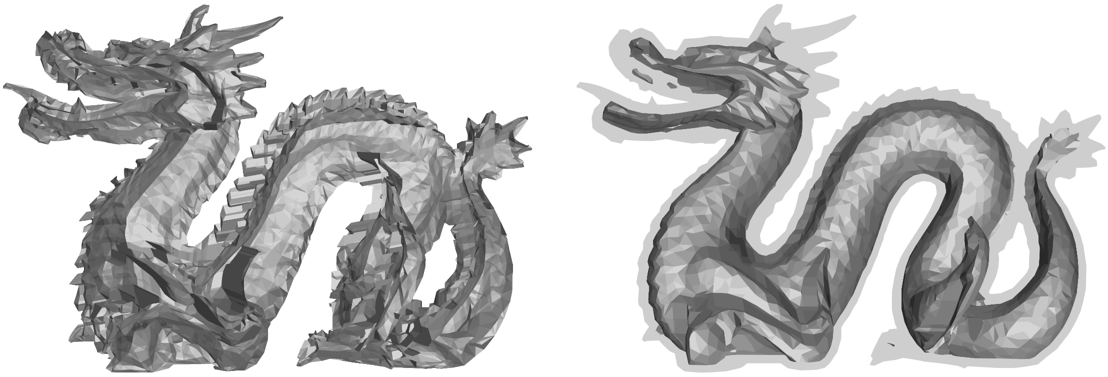
Figure 1.2: The calculation of the lion shape was done with 292 primitives in both cases. The medial axis of the polygonal approximation contains numerous artefacts, while the medial axis of the biarc boundary approximation is topologically correct.

randomized divide-and-conquer algorithm runs in expected time $\mathcal{O}(n \log n)$ if mild assumptions on the graph diameter of the medial axis are met. A high-level description, including a formal runtime and data volume analysis, and a proof of convergence (medial axis stability) are given in [5]. The theoretical foundations being laid, we will concentrate on practical and experimental aspects of the algorithm.

First, a careful description of the main medial axis algorithm is given, followed by enhancements ensuring robustness in the presence of geometric degeneracies. This includes (but is not restricted to) the proper classification and treatment of base cases, in order to establish correctness and to gain running speed, for both smooth and non-smooth circular boundary spline inputs. At the end of this section implementation details, experimental data, and selected examples are provided.

In Section 2.3, the class of shapes which can be processed for medial axis computation is extended to domains with holes. By considering the free-form holes as sites, the concept is conveniently adaptable to Voronoi computation (published as [1]). Up to this point, divide-and-conquer for Voronoi computation has had much attention from a theoretical point of view, but was no favorite for implementation, neither for point sites [65], nor for more complex sites as lines segments, circles and circular arcs [37, 73, 83, 86]. Problems that may arise include complicated merge steps and bisector intersection, as well as the resulting expensive discarding and recomputing of parts of the diagram. An alternative strategy is incremental insertion [74], which turns out to be nontrivial for general sites, as either the splitting into simpler pieces [10], or the a priori insertion of reference data [62] becomes necessary. What all these algorithms have in common is that the bisector curves take part in the computation, which, already in the case of line segments, may be composed of up to 7 pieces, and may even be two-dimensional if not defined carefully in the case of shared endpoints.

The approach proposed in Section 2.3 shifts the problematic part to the divide step, by first extracting the combinatorial structure of the Voronoi diagram, and filling in the bisector curves later on (if they are needed at all). The shape with holes (representing a domain and Voronoi sites) is transformed into a so-called *augmented domain*, which is combinatorially simply connected. This augmented domain has a combinatorially tree-like medial axis, and



(a) Medial axis (b) Trimmed offset

Figure 1.3: Piece-wise linear approximation of the Euclidean medial axis and trimmed offset for a dragon mesh with 12, 000 faces, using a quasi-metric defined by a tetrahedral unit ball.

can as such be processed by the medial axis algorithm introduced in Section 2.2. This section is completed by some experimental data on sites of varying complexities.

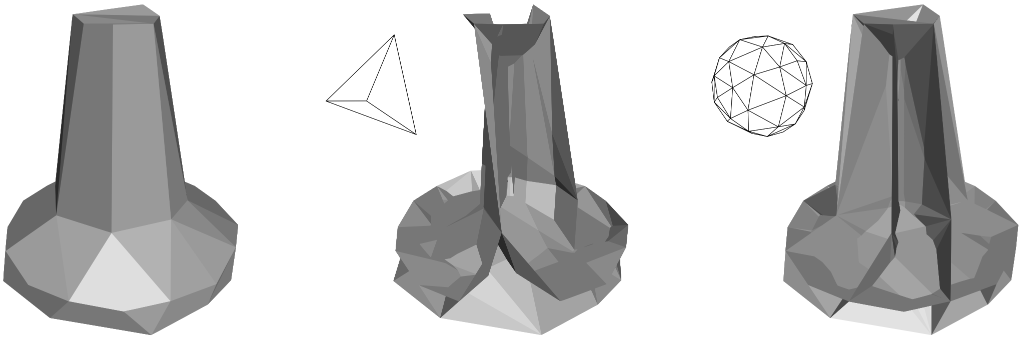
In Section 2.5 we discuss the possibility of algebraically correct computation of the medial axis and Voronoi diagram, given some requirements on the circular arc boundary (Rational Circular Arc Boundary). It is shown that on such an *RCAB*, the computation of the bisectors as well as of the confining endpoints of the medial axis edges can be performed over the field of $\mathbb{Q}$ with a small number of adjoint square-roots (published as [4]).

In Section 2.4 this chapter is concluded by describing an approach for trimmed offset computation which takes advantage of the combinatorial representation of our medial axis via base cases.

1.1.2 Chapter 3: Piece-wise linear medial axis in 3D

As already mentioned, exact medial axis computation for shapes in space is a topic that has not attracted much attention so far. In fact [33] is to our knowledge the only implementation-oriented work in this context. Problems that occur in this context are on one hand the complex topological structure of the medial axis and on the other hand the high algebraic degree of some axis primitives.

In the second main part of this thesis we present an approach that computes the exact medial axis of a triangulated solid (i.e., a solid object whose boundary surface is a triangular mesh) with respect to a piecewise linear quasi-metric $\delta_{\tilde{B}}$ [85] induced by a convex polyhedral unit ball $\tilde{B}$ (see also Minkowski functionals [70]) as introduced in [3]. While the use of more general convex distance functions for bisector and Voronoi computation is no novelty in theory [26, 67], these generalized distances have not been used much in practice so far, and not for medial axis computations in particular. This is quite surprising, considering that for given rational data (rational coordinates of mesh and unit ball vertices) the resulting linearity of the structure allows all computations to be performed within the field of rational numbers. We take advantage of this, providing a robust and stable implementation of the algorithm. Practical results and details are collected in Section 3.3.4.



(a) Tower mesh (b) Simple unit ball (c) Large unit ball

Figure 1.4: Tower mesh and medial axis approximations with respect to two different polyhedral unit balls (4 faces and 128 faces).

The quasi-metric $\delta_{\tilde{B}}$ induces a piece-wise linear medial axis transform $\mathrm{MAT}_{\tilde{B}}(\Omega)$, which describes the shape Ω fully and exactly, see Figure 1.3a for an example. In order to deal with the structural complexity of the medial axis in 3D, single components are computed via their linear projections on the boundary contacts defined by components of special maximal* balls[1] and the boundary of Ω (see also Section 3.3.1). In this way the problem of medial axis construction in 3D is reduced to a number of two-dimensional problems, namely the computation of so-called *contact arrangements* (see Section 3.3.2 for a detailed description of the algorithm).

The use of polyhedral unit balls now does not only turn the resulting medial axis into a piece-wise linear structure, but also permits interesting operations such as implicit pruning, resulting in pseudo-seams which will be introduced in Section 3.1. This allows us to influence the structure and complexity of the medial axis by varying combinatorial and geometrical properties of the unit ball. Figure 1.4b and Figure 1.4c show how strongly axes may differ for different polygonal unit balls. Also in Section 3.1 we will discuss the occurence of so-called *jump-edges* (on the plane) and *jump-sheets* (in the 3D case), which require the definition of maximal* balls in the first place. Furthermore, it is shown in Section 3.3.3 that the representation via $\mathrm{MAT}_{\tilde{B}}(\Omega)$ is very convenient to compute trimmed offsets with respect to $\delta_{\tilde{B}}$, see Figure 1.3b.

Finally in Section 3.2.1.3 the close relation between the medial axis $\mathrm{MA}(\Omega)$ induced by the Euclidean unit ball and $\mathrm{MA}_{\tilde{B}}(\Omega)$ induced by a piece-wise linear metric will be described. This also identifies $\mathrm{MA}_{\tilde{B}}(\Omega)$ with respect to a piecewise linear metric $\delta_{\tilde{B}}$ as an approximation of the Euclidean medial axis, where the quality of the approximation depends on the chosen unit ball $\tilde{B}$.

1.1.3 Related publications

The content of Section 2.2 is based on the ideas formulated in [7] and developed in [2]. The version in this thesis further elaborates many aspects of these two works and provides additional example and runtime data.

---

[1] The notion of maximal* is explained in Definition 3.1.5 in Section 3.1.

In Section 2.3 and Section 2.4 relevant parts of the concepts formulated in [1] are picked up and again complemented by enhanced examples.

Section 2.5 is based on [4]. Where appropriate, adaptions and extensions with regard to contents and illustrations have been made.

The 2D part of Chapter 3 is almost completely original work. The 3D medial axis computation algorithm treated in Section 3.3 is based on [3]. However, this thesis goes into more detail, ties up possible loose ends and provides in general more information about the structure. In addition a more extensive example study is provided.

1.2 General definitions

In this section we will provide some definitions which are relevant throughout the thesis. First we define the set of shapes, for which a meaningful medial axis can be defined.

Definition 1.2.1. *A shape* $\Omega \subset \mathbb{R}^d$ *with* $d = 2, 3$ *is a connected open set. The boundary* $\partial\Omega$ *is then the closure of* Ω *minus* Ω *itself.*

This definition guarantees that a shape is full-dimensional, and contains no features of lower dimension as e.g flat sheets in space.

Of most importance are of course the introduction of maximal inscribed balls and in consequence of the medial axis.

Definition 1.2.2. *Given a convex distance function* δ *and a central point* O*, the associated unit ball* $\tilde{B} \subset \mathbb{R}^d$ *is the convex open set of points* p*, for which*

$$\delta(O, p) < 1 \ .$$

A ball B *is an arbitrarily translated and scaled copy of* $\tilde{B}$*.*

This means that the form of a ball B is directly related to the underlying distance function. If not mentioned otherwise, then it can be assumed that δ is the Euclidean distance, and thus B is an Euclidean ball.

Definition 1.2.3. *Given a shape* Ω*, a ball* $B \subseteq \Omega$ *is called maximal inscribed, if there does not exist a ball* $B' \subseteq \Omega$*,* $B' \neq B$*, which contains* B*. We denote the set of all maximal inscribed balls with* $\mathrm{MAT}(\Omega)$ *(medial axis transform of* Ω*).*

Definition 1.2.4. *A point that is shared by the boundary of a shape* Ω *and the boundary of a maximal inscribed ball* B *is called a* footpoint *of* B *on* $\partial\Omega$*.*

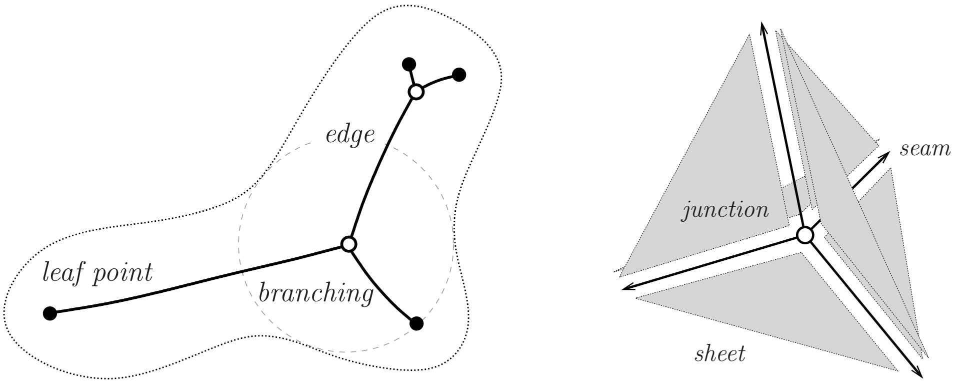
Definition 1.2.5. *Given a shape* Ω*, its medial axis* $\mathrm{MA}(\Omega)$ *is defined as the locus of centers* c_B *of all maximal inscribed balls* B *in* Ω*:*

$$\mathrm{MA}(\Omega) := \{c_B \mid B \in \mathrm{MAT}(\Omega)\} \ .$$

The Hausdorff distance is a measure for the maximal devation between two sets in $\mathbb{R}^d$.

Definition 1.2.6. *For two sets* X *and* Y *in* $\mathbb{R}^d$ *the symmetric Hausdorff distance between these two sets is defined as*

$$\mathrm{HD}(X, Y) = \max(\sup_{x \in X} \inf_{y \in Y} \|x - y\|, \sup_{y \in Y} \inf_{x \in X} \|x - y\|) \ .$$



(a) 2D structure (b) 3D structure

Figure 1.5: The structure of the medial axis in two and three dimensions.

Definition 1.2.7. *Let Ω be a shape in $\mathbb{R}^d$, and $\tilde{B}$ be a unit ball in the same dimension. The trimmed inner offset of Ω at distance ϱ with respect to $\tilde{B}$ is defined as*

$$\Omega^{\varrho} = \Omega \setminus \bigcup_{x \in \partial\Omega} B(x, \varrho)$$

where $B(x, \varrho)$ is a copy of the the unit ball centered at x and scaled by ϱ.

Medial axis structure in two and three dimensions

The medial axis of a shape in the plane is a skeleton-like structure, composed of one-dimensional components (bisector curves) that join at points of degree ≥ 3. The bisector curves are also called *edges*, and their intersections *branching points*. For general shapes with non-degenerate configuration, the branching points have degree 3, meaning that they are defined by the joint of three axis edges and are thus center points of balls with three footpoints.

For a simply connected shape, the medial axis in 2D is a tree. For this reason, points of valency one are called *leaf points* of the medial axis (see Figure 1.5a)[2]. If the shape has holes, then the resulting medial axis contains a cycle for every hole. For more details on pierced shapes see Section 2.3.

In space, the medial axis of a shape is in general composed of two-dimensional components, namely bisector surfaces (for special set-ups, as e.g. a cylinder, also one-dimensional parts may occur). They are called *sheets* of the medial axis. Maximal inscribed balls centered on a *sheet* induce two footpoints. For non-degenerate constellations, three *sheets* intersect in a one-dimensional *seam* (trisectors[3] which are composed of centers of balls with three footpoints), and four *seams* intersect in a so-called *junction point* (see Figure 1.5b). A maximal ball associated with a junction point has usually four footpoints.

---

[2] If the boundary of a shape is not tangent continuous at a point p and if this point induces a convex feature, then an edge of the medial axis goes towards p. Note however that the point p itself is not part of the medial axis in this case, as a maximal inscribed ball is an open set.

[3] Do not confuse with the notion of distance trisector curves as introduced in [13].

For shapes in space which have a connected boundary (note that this also includes objects pierced by tunnels as e.g. a torus), the respective medial axis is also connected and does not enclose any volumes. In this work we do not consider shapes with disconnected boundaries as it occurs for objects which enclose empty bubbles (*Swiss cheese* model).

Chapter 2

Circular boundary representation

2.1 Preliminaries

In this chapter we only consider medial axis computation of shapes in the plane with respect to the Euclidean distance function.

Given a shape Ω as defined in Definition 1.2.1 in $\mathbb{R}^2$, a maximal inscribed Euclidean ball with respect to Ω as defined in Definition 1.2.3 is also called a *maximal disk* for the remainder of this chapter.

Biarc approximation

Biarc approximation of free-form curves has been studied by many authors, see e.g. [52, 79] and the references cited therein. As this is not the topic of this thesis, we will not go into detail here and therefore only provide some basic information about this way of shape approximation.

A biarc is obtained by joining two circular arcs in a way such that they possess a common unit tangent vector at their joint. By this means, smoothness on the boundary of a shape can be maintained. In view of the medial axis computation the arcs and their joints are chosen such that the curvature extrema of the given shape boundary are preserved. This suggests the use of so-called spiral biarcs as introduced by [80], and allows arbitrary close approximation by adaptive bisection (see [2]).

Observation 1. *Consider a shape Ω in the plane with piece-wise smooth boundary $\partial\Omega$. Then for any $\varepsilon > 0$ there exists a spiral biarc approximation $\partial\Omega'$ such that $\mathrm{HD}(\partial\Omega, \partial\Omega') < \varepsilon$ and which preserves the curvature extrema of $\partial\Omega$.*

Given a sequence of approximating curves that converge to the (exact) boundary of a given planar domain, the medial axes of the approximate domains do not necessarily converge to the medial axis of the given domain (e.g. polygonal approximations, where each convex vertex creates its own branch of the medial axis, see Figure 1.2a). However, since the curvature maxima are preserved by the spiral biarc approximation, the number of leaves of the approximate medial axis is equal to the number of leaves of the exact medial axis. Moreover, we have geometric convergence as follows.

Assume that $\mathrm{HD}(\partial\Omega, \partial\Omega') < \varepsilon$ for a shape Ω and an approximation Ω' of it. Then for any $p \in \mathrm{MA}(\Omega)$ which is sufficiently far away from the leaves (where the required distance tends to zero as $\varepsilon \to 0$), it is possible to derive a bound on the distance d_p to the nearest point on

MA(Ω'), namely,

$$d_p \leq \frac{4}{1-\cos(\xi_p/2)} \cdot \varepsilon \ .$$

Here, $\xi_p \in [0,\pi]$ is the maximum angle between any two rays that connect the center of the maximal inscribed circle which is centered at p with any two of its tangency points. Consequently, except for the vicinity of the leaves, the medial axis inherits the approximation order of the boundary approximation by spiral biarcs.

Observation 2. *Consider the boundary $\partial\Omega$ of a shape, and for any $\varepsilon > 0$ the spiral biarc approximation $\partial\Omega'$ with* HD($\partial\Omega, \partial\Omega'$) $< \varepsilon$*, then*

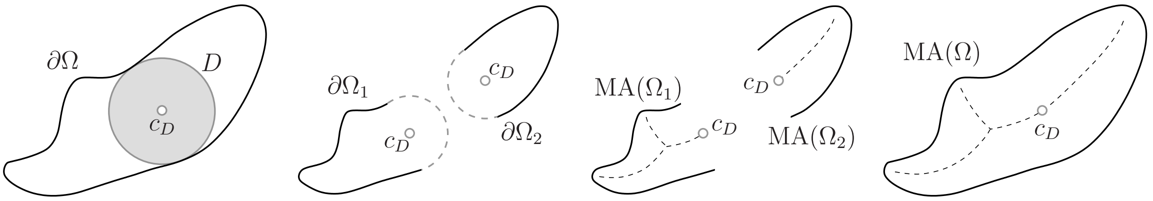
$$\mathrm{MA}(\Omega') \overset{\varepsilon\to 0}{\longrightarrow} \mathrm{MA}(\Omega) \ .$$

If the medial axis of the original shape contains only branchings with valency three, then the spiral biarc approximation preserves the topology of the axis, provided that the error of the boundary approximation is sufficiently small.

For more details, proofs and algorithms of the above we refer to [2, 5]. For the remainder of this chapter we will only consider shapes Ω with spiral biarc, or even more general, circular arc representations.

Definition 2.1.1. *For the remainder of this chapter we denote with Ω a simply connected shape with a circular arc representation, meaning that $\partial\Omega$ consists of a finite set of circular arcs (and possibly line segments). Furthermore, two neighbored arcs on the boundary are considered to lie on different supporting circles, otherwise they are regarded as a single arc.*

Definition 2.1.2. *If all transitions between the arcs of $\partial\Omega$ are tangent continuous, then $\partial\Omega$ is called a* smooth *or C^1 boundary. Otherwise, a boundary that contains non-continuous arc joints, defining either reflex or convex corners, is called a non-smooth or C^0 boundary.*



(a) Maximal disk (b) Divide (c) Compute axes (d) Conquer

Figure 2.1: If the shape Ω is decomposed via a maximal disk D, then the partial axes of the two resulting subshapes Ω_1 and Ω_2 can be computed separately. The axis of Ω is the concatenation of $\mathrm{MA}(\Omega_1)$ and $\mathrm{MA}(\Omega_2)$; shown here from left to right.

2.2 Medial axis of simple shapes

In this section we provide a description (and details of the implementation) of a randomized divide-and-conquer algorithm which computes the exact medial axis of any simply connected shape Ω with a circular arc representation. If the boundary $\partial\Omega$ consists of n arcs, then the expected runtime is $\mathcal{O}(n \log n)$ under the assumption that the graph diameter of the medial axis is $\Theta(n)$. Even for shapes with heavily branched medial axes this only plays a role for coarse biarc approximations.

Observation 3. *The number of branching points of the medial axis is independent of the input size n (the number of circular arcs on the boundary) which, in turn, grows arbitrarily with the user-defined accuracy of the spiral biarc approximation.*

The algorithm is based on the fact that decomposing a given shape with an inscribed disk leaves two (or more) subdomains whose medial axes can be computed independently (see Figure 2.1a to Figure 2.1d for illustration). This observation has first been made use of in [28]. It holds for simply connected planar shapes of any form, and is particularly suited for circular boundaries as this allows the convenient definition of very simple abort criteria for the divide step.

In a nutshell, the algorithm proceeds as follows.

- *Divide step:* A random maximal disk is computed for a shape, and it is checked wether the induced decomposition is progressive (see Definition 2.2.1). If this is not the case then the disk is recomputed deterministically to fulfill this requirement. Each partial shape is then treated recursively, until one of the base cases is reached, for which the medial axis is calculated directly.

- *Conquer step:* Here only the computed medial axes of the partial shapes are concatenated as they fit together at the centers of the maximal disks, used during the divide step.

By this means the expensive and critical computations are delegated to the divide step, while in the conquer step, the subsolutions are simply glued together without the need of any merging or adjustment operations. This reduces the effect of error accumulation, and keeps numerical imprecision, if it occurs at all, locally restricted.

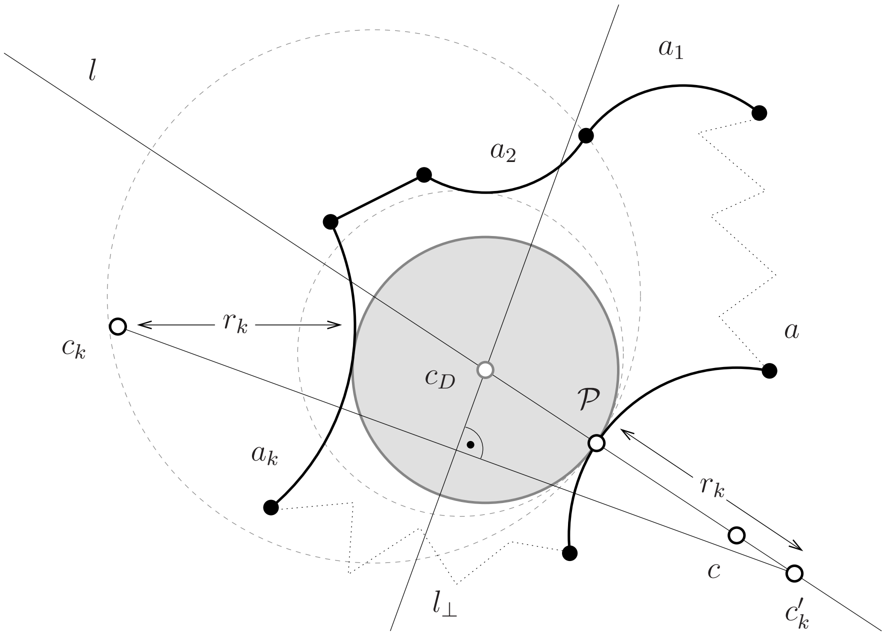


Figure 2.2: Constructing a maximal disk tangent to arc a at point $\mathcal{P}$. The dashed disks induced by a_1 and a_2 overlap other boundary arcs.

2.2.1 Divide step

The divide step carefully chooses a maximal disk D and splits the shape boundary into two or more chains, depending on the number of footpoints (see Definition 1.2.4) of D.

Definition 2.2.1. *A maximal disk is called* dividing *disk, if the boundaries of all partial shapes that result from the induced decomposition are combinatorially smaller (consist of less arcs) than the boundary of the intial shape. Otherwise it is called a* non-dividing *disk.*

The resulting subshapes are completed with circular arcs which have ∂D as their supporting circle. As Ω is defined as the original complete shape, we will henceforth write Ω' for an arbitrary subshape of Ω.

Definition 2.2.2. *Circular arcs on the boundary of subshapes, that have the boundary of dividing disks as their supporting circles are called* auxiliary *arcs, while arcs that are part of $\partial\Omega$ are called* original *arcs.*

Every maximal disk is, via its footpoints, uniquely assigned to two or more arcs on $\partial\Omega$. A possible way to pick a random maximal disk in Ω' [5] is to choose a random original arc a on $\partial\Omega'$ and to construct the maximal disk whose center lies on the line through the center of the supporting circle of a and a fixed point $\mathcal{P}$ on a (e.g., its midpoint). The resulting disk then has $\mathcal{P}$ as one of its footpoints.

Given the set of arcs a_i, $i = 1 \ldots n$, that represent $\partial\Omega'$, this is accomplished by iteratively constructing disks that are tangent to a at the point $\mathcal{P}$ and to some other original arc a_k (for k starting at 1). If the resulting disk still intersects or overlaps an original arc a_l with $1 \leq k \leq l \leq n$, then a new disk (which is smaller than the preceding one) tangent to a_l is computed, until a valid maximal disk D is obtained. For step by step details of the `maxdisk` procedure see Algorithm 1).

Observation 4. *An auxiliary arc lies on the boundary of a maximal disk and thus does not contain a footpoint of any other maximal disk. Therefore only original arcs have to be considered for the iterative construction of a new maximal disk in a partial shape.*

As all n arcs of the boundary have to be checked, an $O(n)$ time complexity for the computation of a single maximal disk is obtained.

Algorithm 1 `maxdisk`(a,$\partial\Omega'$) computes a maximal disk in Ω' tangent to a

```
procedure maxdisk($a$,$\partial\Omega'$)
                                                  ▷ choose footpoint on arc
    if endpoint of $a$ is reflex corner of $\Omega'$ then
        $\mathcal{P} \leftarrow$ reflex endpoint
    else
        $\mathcal{P} \leftarrow$ arbitrary inner point of $a$
    end if
                                   ▷ iterative computation of the maximal disk
    $D \leftarrow$ disk with infinite radius tangent to $a$ at $\mathcal{P}$
    $n \leftarrow$ number of arcs on $\partial\Omega'$
    for $i = 1 \dots n$ do
        $a_i \leftarrow$ i$^{th}$ arc of $\partial\Omega'$
        if $a \neq a_i \wedge a_i$ is original $\wedge\ D \cap a_i \neq \emptyset$ then
            $D \leftarrow$ disk tangent to $a$ at $\mathcal{P}$ and tangent to $a_i$
        end if
    end for
    return $D$
end procedure
```

The central part of this calculation is the geometric construction of a disk which is tangent to an arc a at a fixed point $\mathcal{P}$, and which is arbitrarily tangent to another arc a_k. See Figure 2.2 for an illustration. The point c_D, which is the center of the desired maximal disk, is the matter of interest. This point must lie on the line l through c, the center of the supporting circle of a, and $\mathcal{P}$. If you move from the point $\mathcal{P}$ a distance of length r_k (the radius of a_k) towards c, you arrive at the point c'_k. Together with c_k (the center of the supporting circle of a_k) and c_D this point forms an isosceles triangle. Let l_b be the perpendicular bisector of $\overline{c_k c'_k}$, then the intersection of $l_\perp$ and l gives the point c_D. This construction can, with slight modificiations, be applied to pairs of arcs in arbitrary position. When replacing the circular arc a_k by a line segment, the problem can be reduced to the intersection of the line l with an angle bisector of the line perpendicular to l through $\mathcal{P}$ and the supporting line of the segment.

This disk construction is, together with intersection and overlap checks, the most frequent and numerically most complex step in the entire medial axis algorithm. Thus the main atomic operations are computing intersections of circles and lines.

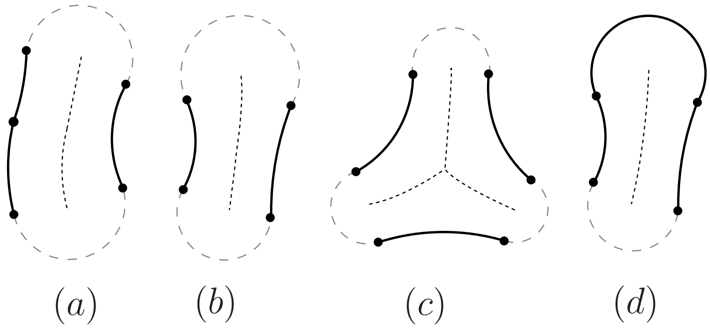


Figure 2.3: Base cases for C^1 boundaries.

Base cases

We will now provide a classification and analysis of appropriate termination conditions for the divide step. When considering a C^1 boundary as precondition, then any shape bounded by circular arcs and line segments can be decomposed into only four base cases; see Figure 2.3.

This is simply accomplished by dividing iteratively until the number of original arcs drops below four. Let us argue that the cases in Figure 2.3 cover all possibilities. Observe first that no consecutive auxiliary arcs may occur, because for smooth boundaries we construct every maximal disk at an inner point of an arc a.

- All possible combinatorial constellations with 3 original arcs are covered in the cases (a), (c), and (d), provided no consecutive auxiliary arcs are allowed.
- The combination shown in case (b) is the only one which may occur with 2 original arcs. A base case with two *consecutive* original arcs, connected by an auxiliary arc while guaranteeing smoothness at all vertices would require all arcs to be on the same supporting circle, what is considered to be a degenerate case (see Definition 2.1.1).
- A base case with one original and one auxiliary arc would also require both arcs to be on the same supporting circle. This is not a real base case, however, it can be used to speed up the decomposition process as will be shown in Section 2.2.2.

If reflex and convex vertices on the boundary are allowed, then more attention has to be paid to the choice of the point $\mathcal{P}$ to keep the number of arising base cases small. If a randomly chosen arc a has some reflex endpoint, we do not choose its midpoint but rather the reflex endpoint itself as $\mathcal{P}$.

Furthermore the termination conditions have to be slightly extended. We keep on splitting until all of the following criteria are satisfied:

1. The number of original arcs is ≤ 3.
2. There exists no original arc with a reflex vertex.
3. If three original arcs are consecutive then no convex vertex occurs.

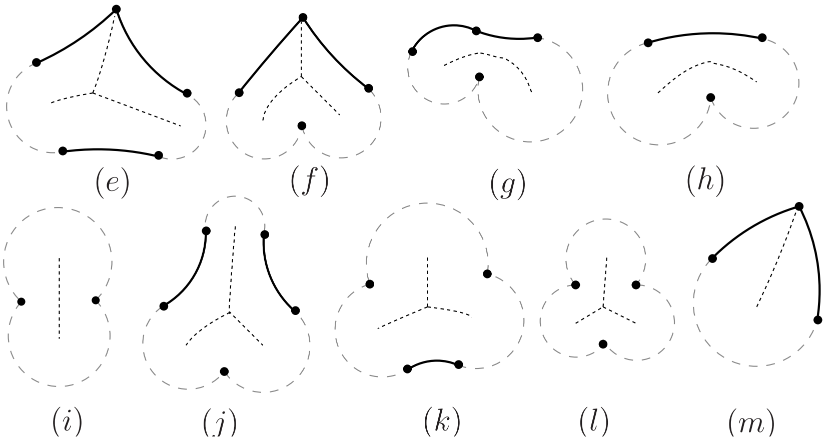


Figure 2.4: Additional base cases for C^0 boundaries.

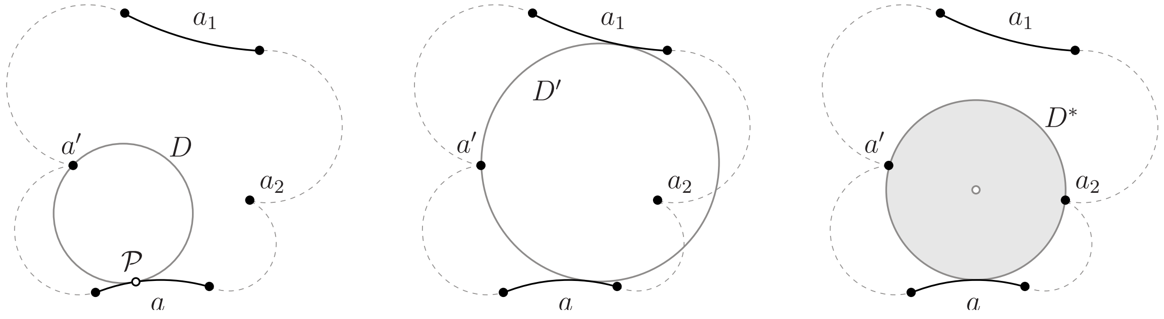
This results in nine additional possible base cases as shown in Figure 2.4. These new cases cover all possible non-smooth varations of the cases (a), (b) and (c). Variations include the turning of a smooth vertex into a convex one and the replacement of an isolated original arc with a reflex vertex as induced by the termination criteria. The base case (d) has no non-smooth derivatives because of splitting rule 3. Additionally, if we consider a reflex vertex as an arc with length zero, we can maintain the observation that no consecutive auxiliary arcs do occur. Together with the following analytic enumeration of the new base cases it is obvious that the arguments concerning completeness of the smooth cases apply to the situation of a C^0 boundary as well.

- For smooth case (a) the joint vertex can become convex, the isolated original arc can be exchanged by a reflex vertex, or both. These variations are covered by the cases (e), (g), and (f) in Figure 2.4.
- The two variations of smooth case (b) are obtained by replacing either one or both original arcs by reflex vertices. See case (h) and case (i) for a realization of this.
- The new base cases (j), (k), and (l) represent all possible combinatorial variations of smooth case (c) caused by turning isolated original arcs into reflex vertices.
- Finally, the case (m) can be derived from the earlier mentioned, for smooth boundaries degenerate, case of two consecutive original arcs and a single auxiliary one.

2.2.2 Termination arguments

2.2.2.1 Preventing redundant cuts

During the division process–especially when dealing with reflex boundary vertices–situations may occur where a maximal disk obtained by the `maxdisk` algorithm is not a dividing disk as defined in Definition 2.2.1 and thus fails to decompose the shape into partial shapes with boundaries that are combinatorially smaller. As the property for the partial boundaries to shrink is needed to assure a termination of the algorithm, such a situation may lead to an



(a) Non-dividing disk D (b) Starting disk (c) Branching disk D^*

Figure 2.5: The disk D^*, tangent to three boundary arcs, is constructed by `maxdisk`* after detection of a non-dividing maximal disk D.

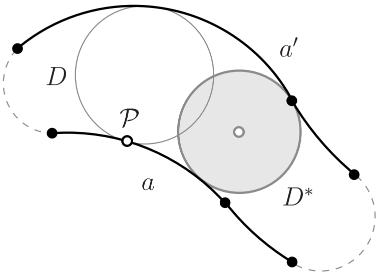


Figure 2.6: The algorithm `maxdisk` for the original arc a finds the maximal but non-dividing disk D which induces the redundant cut splitting into a base case of type (b) and a subshape of the same combinatorial size as the initial one. In the smooth case, as here, the center of the disk D^*, computed by `maxdisk`*, does not have to be a branching point of the medial axis.

infinite loop if a non-dividing disk is returned by `maxdisk`$(a,\partial\Omega')$ for unfavorable choice of a. See Figure 2.5a for an example, where, even independent of the choice of a and $\mathcal{P}$, the constructed maximal disk D causes base cases of the form (h) to be cut away over and over again. The other remaining subshape has the same number of arcs as the preceding one, and so undergoes no combinatorial reduction.

These *redundant cuts* can be detected, and subsequently be avoided, by a more sophisticated choice of the dividing disk. As a pleasing side effect, this choice will also handle the intriguing case of multi-branching of the medial axis.

As soon as a non-dividing disk D, as shown in Figure 2.5a or Figure 2.6 is detected, a new algorithm `maxdisk`* is invoked which computes a disk D^* that is tangent to three arcs on the shapes boundary instead of only two. Similar to the original algorithm `maxdisk`, the procedure traverses all original boundary arcs a_i. It checks, however, which of them is a potential third tangent arc, as the first two are known to be the footarc a and the arc a' chosen by the `maxdisk` algorithm holding the second footpoint of D.

A disk D' induced by three arcs a, a' and a_i now has to satisfy two criteria to be the sought-after dividing disk:

1. Consider the line defined by the two footpoints of the disk D on a and a'. Then the center point of the disk D' must lie on the side opposite to the auxiliary arc that connects a and a'.

2. The disk D' does not overlap any other boundary arc of Ω'.

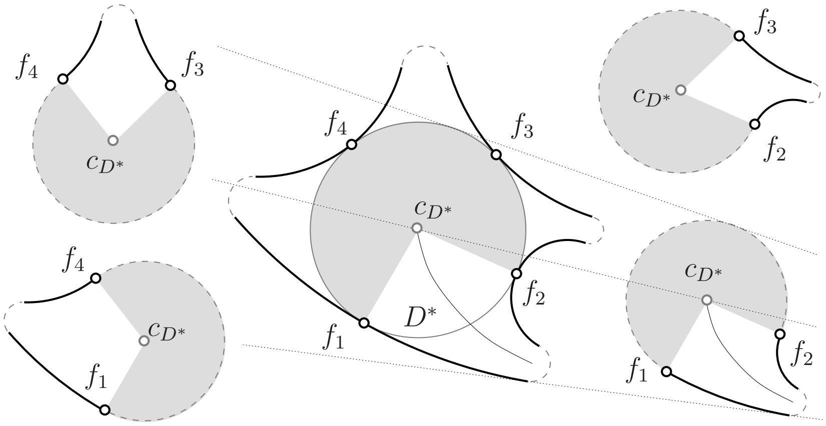


Figure 2.7: A degenerate case where four branches of the medial axis meet at a single point c_{D^*}. The shape is decomposed into four subshapes. This situation is handled by `maxdisk`*.

In the example depicted in Figure 2.5b, the disk D' tangent to the three arcs a, a', and a_1 gets constructed first, which satisfies criterion 1. It is thus a valid starting disk for the following procedure: It is checked if this disk overlaps another arc of $\partial\Omega'$ (here e.g. a_2). The disk being tangent to a, a' and this very arc is constructed and assigned to D'. This is repeated until the obtained disk does not overlap any other part of the shapes boundary, and the process terminates with the desired dividing disk D^*. Note that by this means all intermediate disks satisfy criterion 1. In Figure 2.5c each of the three resulting subshapes boundaries is lacking at least one original arc, namely, one of the tangent primitives. Thus a reduction is guaranteed.

The situation depicted in Figure 2.6 also shows a choice of a and $\mathcal{P}$ which causes a non-dividing disk. Here the center point of the valid dividing disk D^* that gets computed by `maxdisk`*, however, is not a branching point of the medial axis. The reason for this is the smooth transition between arc a' and its neighbored original one.

Note also that in this, as in most cases, another choice of a and $\mathcal{P}$ might render the application of `maxdisk`* unnecessary. The reason why this does not apply to the situation as depicted in Figure 2.5a, is that zero-length arcs representing the reflex corners are not suitable for maximal disk construction by `maxdisk`.

2.2.2.2 Special geometric set-ups

The `maxdisk`* algorithm also recognizes and handles multi-branchings, i.e., nodes of the medial axis with valency four or more. If a valid branching point disk D^* is tangent (or, for the implementation, ε-tangent for a predefined small ε) to $m \geq 4$ primitives, then such a multiple branching point occurs. Every tangent arc defines a point of tangency for D^* on the shape boundary, and the shape is divided into m subshapes which are all joined together at D^*. Figure 2.7 gives an illustration.

When several reflex vertices agglomerate in a relatively small area of a shape (perhaps with no separating boundary parts) then another non-reducable case may occur: a subshape consisting of an arbitrary number of auxiliary arcs, separated by arcs of zero length (as they result from reflex boundary vertices). The medial axis of such a case is a subset of the standard Voronoi diagram, with the zero length arcs as the defining points. With a construction very

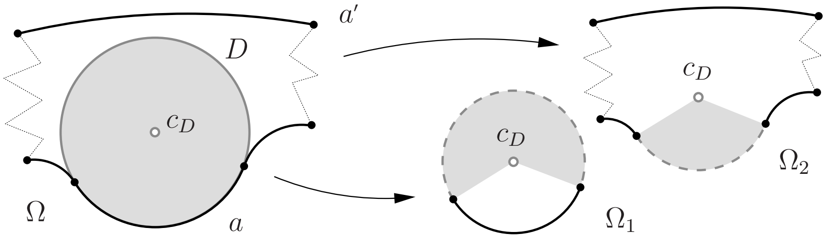


Figure 2.8: The maximal disk found by `maxdisk` for an arc a which defines a locally maximal curvature can be the supporting circle of a itself.

similar to the `maxdisk`* algorithm, these cases can be reduced to base cases of the form (l) from Figure 2.4. Two zero-length arcs (points) neighboured on the subshape's boundary are fixed. A third zero-length arc is then determined in the iterative process, such that the disk defined by these three points does not contain any other point. This disk is a valid maximal disk, which is tangent to the boundary at three points.

2.2.2.3 Local maximal curvature

As said in Section 2.1, the spiral biarc approximation preserves the curvature extrema of the original shape boundary. All leaves of the medial axis are defined by locally maximal curvature regions of the boundary (locally convex features of the shape), but not every locally maximal curvature does necessarily induce a leaf.

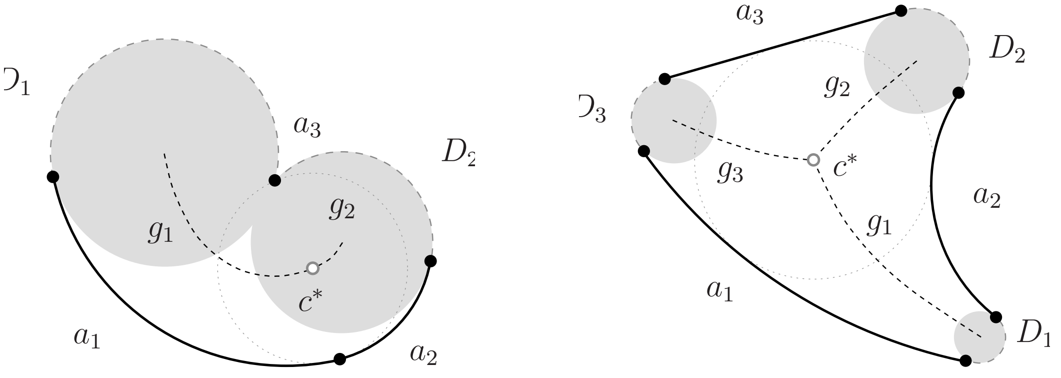
Observation 5. *An arc a of the circular boundary representation $\partial\Omega$ defines a leaf of the medial axis* $\mathrm{MA}(\Omega)$ *if the disk bounded by its supporting circle is a subset of* Ω.

See Figure 2.8, where the arc a represents a locally maximal curvature feature. If the arc a' would intersect ∂D, then no leaf point of the medial axis would occur here. As can be seen, if such a leaf-point defining arc a is chosen for `maxdisk`, then the resulting maximal disk is the one bounded by the supporting circle of a. To achieve a combinatorial reduction, the initial shape is split up as shown in Figure 2.8, where the shape defining solely a leaf point of the axis, is considered to be a base case. Note, however, that by choosing a and $\mathcal{P}$ differently, it is always possible to represent a leaf point by a base case of the type (d) as shown in Figure 2.3.

2.2.3 Conquer step

In the conquer step, the (partial) medial axes of the base cases are computed directly, and then are concatenated at centers of maximal disks whose boundary circles support the respective artifical arcs. At this point, we know exactly which parts of the (global) medial axis correspond to which parts of the boundary of the shape. As the shape boundary is piecewise circular, the medial axis consists of conic arcs. Each such arc is assigned to two primitives on the boundary where it is equidistant from. Possible primitives are circular arcs, line segments, and points (boundary vertices). Different pairs of primitives result in different types of conics:

- Two circular arcs may define an elliptic or a hyperbolic arc, depending on the position of the two supporting circles, and the orientation of the arcs on the boundary. See Section 2.5 and [7] for a more detailed classification.



(a)

(b)

Figure 2.9: The medial axis of a case of type (g) and a case of type (c) in detail.

- A circular arc and a line always define a parabolic arc.
- A circular arc and a point define an elliptic arc if the point lies inside the arc's supporting disk, and a hyperbolic arc, otherwise.
- Two line segments define a straight line.
- A line segment and a point define a parabolic arc.
- Two points again define a straight line.

Let us give two examples for illustratory reasons. First consider a shape of the base case type (g) with a labeling as in Figure 2.9a. The original arc a_1 and the reflex point (or arc of length zero) a_3 define the conic arc g_1. As the point a_3 lies inside the supporting circle of a_1, the curve g_1, leading from the center of D_1 to the center c^* of the disk being tangent to all three non-auxiliary arcs on the boundary, is an elliptic arc. The same applies to the curve g_2, which is defined by a_2 and a_3, and goes from c^* to the center of D_2. Note that c^* is not a branching point of the medial axis, as there is a smooth transition between a_1 and a_2.

Next, consider a case of type (c) where a branching of the medial axis occurs (Figure 2.9b). Curve g_1 is an elliptic arc defined by a_1 and a_2. The curves g_2 and g_3 are parabolic arcs, as one of the defining arcs (a_3 in this case) is a line segment, respectively. The special feature of this base case is the branching point c^* which is the center of a tritangent maximal disk, and at which all three aforementioned curves join. As said in Section 1.2, a branching point is a point on the medial axis which is equidistant from at least three primitives on the boundary. Its assigned maximal disk touches the boundary at more than two points. The positions of the branching points are needed as endpoints for our conic arcs, and thus have to be computed directly. In this example, the point c^* has to be computed, that has the same distance to the two arcs a_1 and a_2 and the line segment a_3.

Branching points

If we consider a base case of type (c), with all possible combinations of circular arcs and line segments and the possibility to replace arcs by (reflex) points, as in the base cases (j), (k), and (l), we get ten possible combinations of three primitives. What we are looking for is the disk that is simultaneously tangent to all three of them. This is a long-known geometric issue, also known as the Apollonius problem, named after the ancient Greek geometer who posed this classical problem. (discussed among others by [55, 75]). As up to eight circles may satisfy the tangency conditions to the circles (lines) supporting the primitives, we have the problem of singling out the unique valid disk that touches them at the right portion. This task has been implemented for all triples of primitives, as this is needed in the computation of all the branching points ocurring in the base cases (c), (e), (f), (j), (k), and (l). For more details see Section 2.5 and [7].

2.2.4 Putting the pieces in place

By combining the procedures introduced above we obtain the main algorithm for the medial axis computation, as lined out in Algorithm 2. Its input is the shape Ω, represented by its piecewise circular boundary $\partial\Omega$. The algorithm divides Ω recursively into partial shapes, until they match any of the base cases introduced before. The choice of the disk (constructed by `maxdisk`) which is used for the decomposition is random at first.[1] If a non-dividing maximal disk occurs, then a disk centered at a branching point is computed by the extended algorithm `maxdisk`*. If the state of a base case is reached, there are two possible ways to proceed:

- The medial axis of the base case is computed directly. It exclusively consists of conic arcs. This is one of the benefits from the circular boundary representation. For algebraically exact computation of the arcs see Section 2.5.
- For certain applications, the curve equations of the axis segments may be of small or no interest at all, as rather the topological or combinatorial structure is needed. Through the use of base cases, which reveal various special features of the shape and its medial axis (branching points, local curvature maxima, etc.), it is easy to derive useful information on the axis without calculating the conic arcs right away. By storing the combinatorics of all base cases, the exact medial axis can be computed at a later point, and for any required part of the shape.

Expected runtime $\mathcal{O}(n \log n)$

The number of branching points of the medial axis of a circular boundary representation is independent from the accuracy of the spiral biarc approximation and thus the number n of approximating arcs. By this means, for large enough accuracy and n, the graph diameter of the medial axis is $\Theta(n)$.

Corollary 1. *If the boundary representation $\partial\Omega$ consists of n circular arcs, and the medial axis* $\mathrm{MA}(\Omega)$ *has a diameter of* $\Theta(n)$*, then* `medax`(Ω) *computes this medial axis in expected* $\mathcal{O}(n \log n)$ *time.*

Under the assumption that the graph diameter of $\mathrm{MA}(\Omega)$ is linear in n, the center point of a random dividing disk computed by `maxdisk`$(a,\partial\Omega)$ for a randomly chosen boundary arc

---

[1] Other approaches to achieve a more balanced decomposition as discussed in [5] give stronger theoretical bounds and might prove useful on multi-processor architectures. However on single CPU architectures, using solely random choices has turned out to be more efficient [7].

Algorithm 2 `medax`(Ω') computes the medial axis of Ω'

```
procedure medax($\Omega'$)
                                        ▷ decide if shape is further decomposed
    if $\Omega'$ is base case then
        compute medial axis of $\Omega'$ directly
    else
        $a \leftarrow$ random original arc in $\partial\Omega'$
        $D \leftarrow$ maxdisk($a$, $\partial\Omega'$)
        if $D$ is non-dividing disk with second footpoint on $a'$ then
            $D \leftarrow$ maxdisk*($a$, $a'$, $\partial\Omega'$)
        end if
        $k \leftarrow$ number of footpoints of $D$
        split $\Omega'$ into $\Omega'_1, \ldots, \Omega'_k$
                                        ▷ recursive medial axis computation
        for $i = 1 \ldots k$ do
            medax($\Omega'_i$)
        end for
    end if
end procedure
```

a lies on the diameter with constant probability. Thus MA(Ω) is split at this center point into two parts of expected size $\Theta(n)$. A randomized runtime of $\mathcal{O}(n \log n)$ results.

On the other hand, if the diameter is rather small ($\Theta(\log n)$ for a very ramified shape), it is very likely that a randomly chosen arc induces a dividing disk, whose center lies rather far from the center of the graph. This is also confirmed in the practical results provided in Section 2.2.5.

2.2.5 Implementation and examples

2.2.5.1 Implementation with CGAL

The algorithm presented in the previous section has been implemented in C++ for matters of performance and availability of supporting libraries. As many geometrical constructions and checks are necessary during the course of the algorithm, the Computational Geometry Algorithms Library (CGAL) [90] proved to be the most appropriate choice. CGAL is a C++ package for combinatorial, algorithmic, and geometrical solutions with an emphasis on flexibility, stability, exactness, and performance. It provides simple geometric calculations as intersection, position, and distance checks, supports the visual output with simple GUIs and visualization libraries as Qt [89], and also incoorporates useful packages of the Boost library [87].

The main benefit of CGAL is, however, the possibility to choose between different kernels (providing different number types) which satisfy the demanded requirements, and which may be varied with minimal effort due to CGAL's template architecture. The implementation of the medial axis algorithm has been realized in two different versions:

1. To achieve an implementation as reliable as possible, the exact rational number type *Gmpq* from the GNU Multiple Precision Arithmetic Library [91] has been chosen in the

first version. The main reason for this decision is the representation of a circle as an algebraic quadratic equation in CGAL. An arbitrary point on a circle is a solution of this equation, and thus has irrational coordinates, in general. As float numbers then are necessarily imprecise, we seek rational points which exactly lie on a circle defined by three rational points. It is known that such a circle has the following properties:

- The center of the circle has rational coordinates.
- Points with rational coordinates lie dense on the circle.

So it is possible to find a rational point as near to any point on a circle as desired. This has been implemented in our program following the results from [21]. Due to the very large integers needed in these calculations, the choice of an elaborate rational type as *Gmpq* is inevitable for a reliable implementation.

2. If exactness is not the main issue (and, as observed in practical tests, the results do often not decisively differ) then the use of a float number type results in faster runtimes. A version of the program which uses explicitely *double* numbers has been implemented for this purpose. As the statistical evaluation in Section 2.2.5.2 will show, the gain in runtime is considerable, but computational inaccuracy may possibly result in incorrect (though locally restricted) partial solutions.

Problems with the *double* implementation arise especially when dealing with very large circles, which result from three almost collinear points defining an arc. Lines which are nearly parallel also raise a problem, as the resulting intersection point can often not be properly represented by a float type. If such situations occur, the *double* implementation reaches its limitations, and locally incorrect sets of maximal disks are the outcome.

The implementation of the algorithms offer several features for the manipulation of both, the input and the output. Using some of them is occasionally necessary to generate appropriate data, others can be seen as a possibility to experiment with the problem:

- As every considered boundary is a collection of arcs and line segments, and does not consist of one single differentiable function, it does make sense to take a closer look at the connecting vertices between two arcs. The spiral biarcs approximation generally assures a smooth boundary, but as the representation is not totally exact, it makes sense to introduce a small error constant. Via this constant it is decided whether a vertex defines a (convex or reflex) corner of the shape, or if the shape is considered smooth in the neighborhood of this vertex. The constant can be varied to fit the quality of the used input data. The issue of an algebraically exact C^1 circular arc boundary is dealt with in Section 2.5.

- The output of the computed circular boundary representation and its medial axis is realized in two different ways. On one hand, the Qt library from Trolltech [89] is used for the visualization on screen, supporting various functions as translation and zoom. On the other hand, it is possible to write the obtained medial axis directly to PostScript, where the conic arcs are approximated either simply by line segments or by cubic Bézier curves, which are more convenient for output in PostScript than conics.

- There exist various other possible modifications to tune the input or the ouput, as for example the possibility to convert the input arcs into x-monotone arcs before processing, or the use of a bound flag for the arc's radii which causes arcs defined by almost collinear points to be recognized as line segments (which makes sense to avoid numerical errors especially when working with the *double* kernel).

In the following we report on the experimental behavior of the medax algorithm, and display and interpret the produced output for selected examples. We will also include runtimes for offset computation, which will be introduced in more detail in Section 2.4.

With growing approximation quality of the boundary, the computed medial axis converges to the exact axis of the original shape. To rate the influence of the approximation accuracy on the speed of the implementation, several different circular boundary representations of a particular shape have been generated.

Every example is provided with a picture of the original free-form shape together with its medial axis, as well as a detailed view (including an offset) of a smaller fraction. The runtimes in the respective tables are given in seconds, for the different approximations of the shapes. The column `# arcs` gives the number of boundary arcs of a single approximation. For the *double* as well as for the *Gmpq* kernel, `Divide` shows the elapsed seconds for the decomposition into the base cases, `Conquer` for the assembling of the base cases medial axes, adding up to the values in the column `total`. For the *double* kernel also the time needed for the computation of a trimmed offset (see Section 2.4), assuming a complete medial axis, is given. All these values have been averaged over 10 runs.

The calculation of the coefficients for the equations of the conics building the medial axis is included in the times given for the base case computation. It is interesting to see that the amount of elapsed seconds grows in an almost linear fashion with regard to the number of arcs. To emphasize this, the log-log graphs representing the runtimes for the two different kernels are shown with linear reference functions for hypothetical runtimes of 50 μs, 200 μs, 5 ms and 15 ms per arc from the bottom up.

Note again that, by construction, the number of branching points of the medial axis stays the same for all approximations, because the number of leaves is the same as for the original free-form shape.

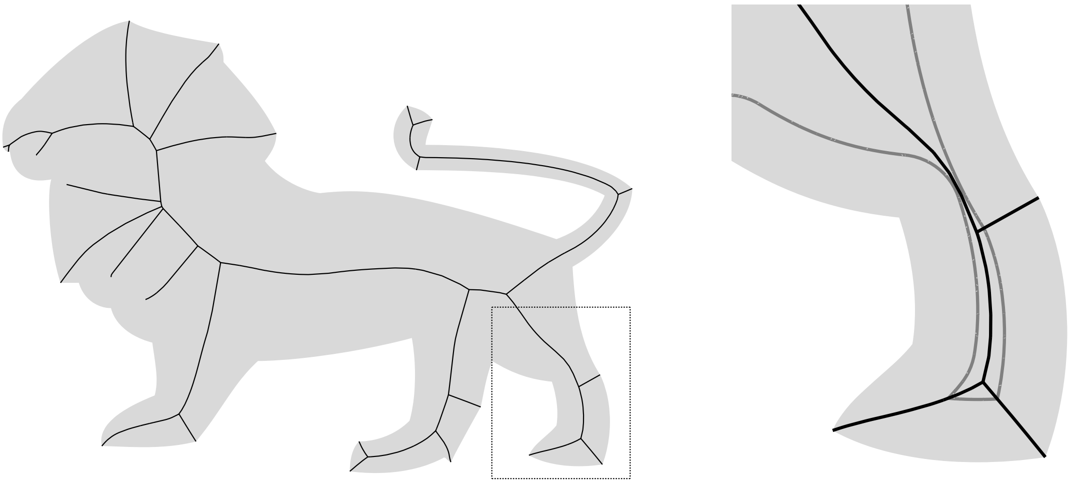
The configuration used for all tests is a 32 bit installation of Linux Debian on an Intel Core 2 Quad Q9300 architecture[2] with 8 GB RAM.[3] As no parallel processing is implemented yet, only one core is used so far.

---

[2] Note that this is a system different to the one used in [2], leading (mainly due to the change from 64 bit to 32 bit for reasons of better comparability) to slightly higher runtimes.

[3] Memory consumption is not a real issue, as the space complexity is also expected $\mathcal{O}(n \log n)$ for $\partial\Omega$ composed of n arcs.

2.2.5.2 Examples

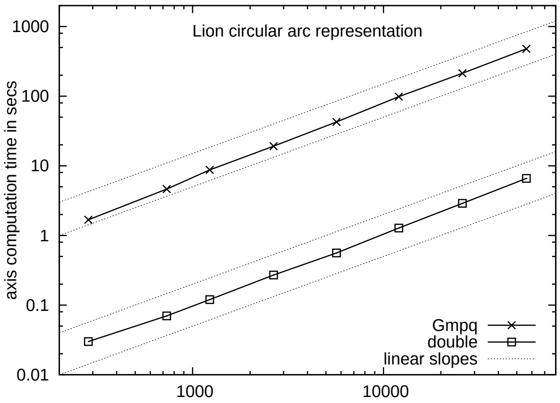


(a) Shape with axis

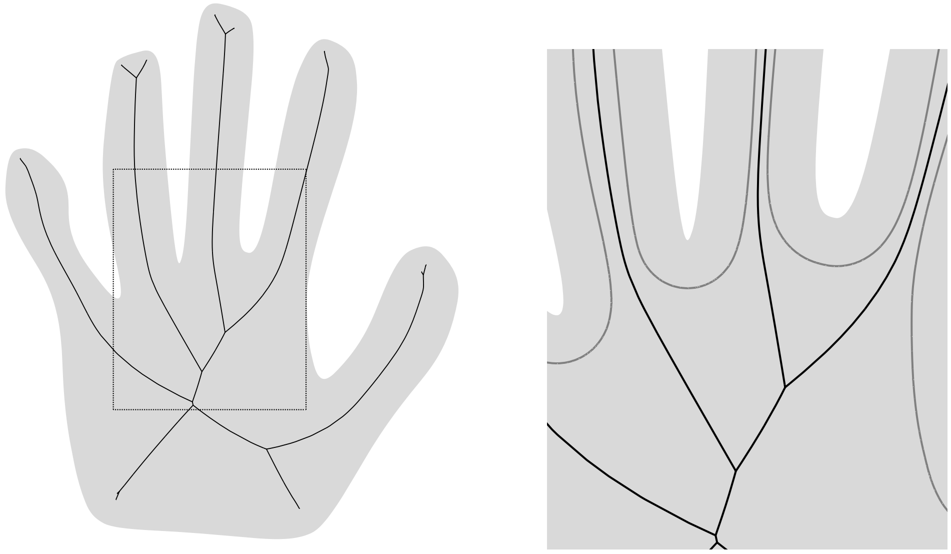
(b) Detail with offset

Figure 2.10: Lion shape

	double				Gmpq		
`# arcs`	`Divide`	`Conquer`	`total`	`Offset`	`Divide`	`Conquer`	`total`
284	0.03	0.00	0.03	0.01	1.58	0.10	1.68
732	0.07	0.01	0.07	0.02	4.43	0.21	4.64
1230	0.11	0.01	0.12	0.03	8.42	0.32	8.75
2656	0.24	0.03	0.27	0.06	18.53	0.62	19.15
5678	0.51	0.05	0.56	0.13	41.36	1.24	42.60
12044	1.20	0.08	1.28	0.28	95.80	2.54	98.34
25948	2.67	0.22	2.89	0.60	208.08	5.32	213.40
56150	6.13	0.47	6.60	1.31	466.70	11.12	477.83



The above table shows the computation times in seconds, needed by the medial axis algorithm for the `Divide` (decomposition) and the `Conquer` (base case axis computation) step, when processing different circular arc approximations of the lion shape in Figure 2.10. The four dotted lines in the associated log-log graph to the left show linear reference functions for hypothetical runtimes of 50 μs, 200 μs, 5 ms and 15 ms per arc from the bottom up.

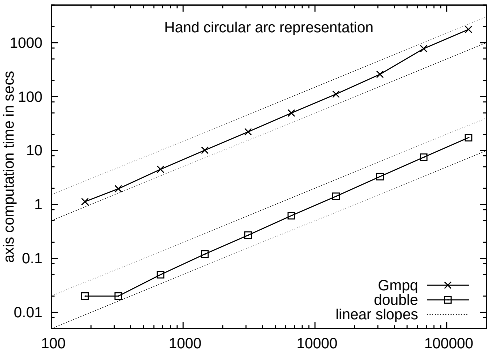


(a) Shape with axis

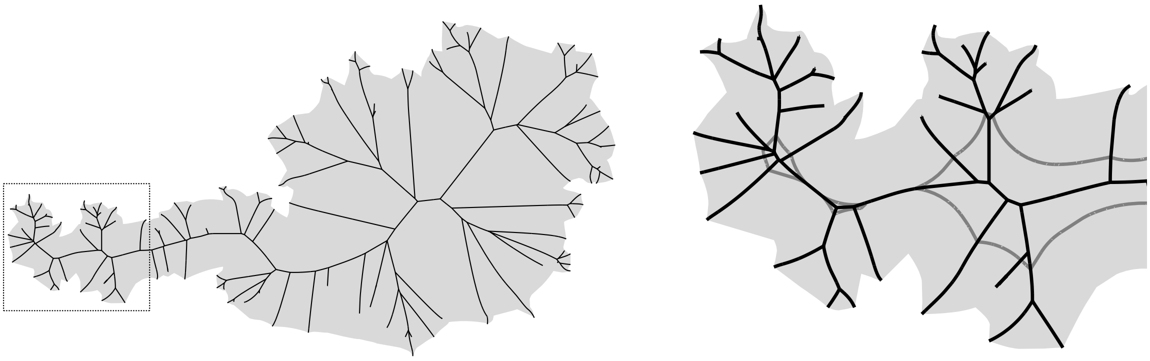
(b) Detail with offset

Figure 2.11: Hand shape

	double				**Gmpq**		
`# arcs`	`Divide`	`Conquer`	`total`	`Offset`	`Divide`	`Conquer`	`total`
180	0.01	0.00	0.02	0.00	1.06	0.08	1.13
322	0.02	0.00	0.02	0.01	1.87	0.10	1.96
674	0.05	0.00	0.05	0.02	4.33	0.16	4.49
1460	0.11	0.01	0.12	0.04	9.83	0.33	10.16
3110	0.24	0.03	0.27	0.07	21.66	0.64	22.31
6616	0.58	0.04	0.62	0.15	48.54	1.23	49.77
14426	1.33	0.08	1.42	0.32	108.63	2.62	111.26
31138	3.11	0.18	3.29	0.69	254.70	5.39	260.10
66764	7.10	0.43	7.53	1.47	762.95	14.05	777.00
146116	16.46	0.92	17.38	3.21	1737.09	29.75	1766.84



The hand shape of Figure 2.11 has a smooth boundary, and its medial axis contains rather few branching points. The resulting good runtime behaviour is reflected in the graph by the proximity of the two kernel runtimes to the respective lower linear reference functions. The average recursion depth during decomposition is 24.4 for the combinatorially smallest boundary and 53.8 for the largest one.

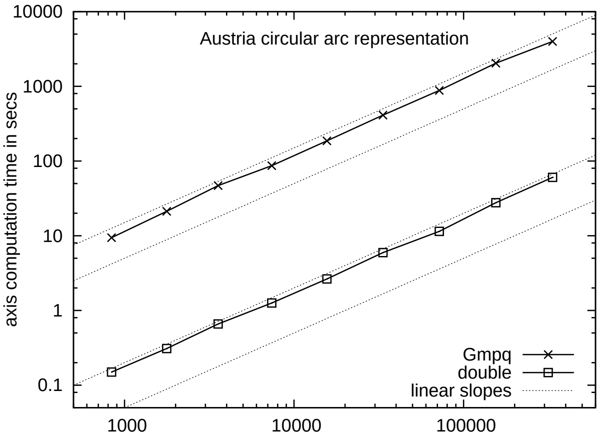


(a) Shape with axis

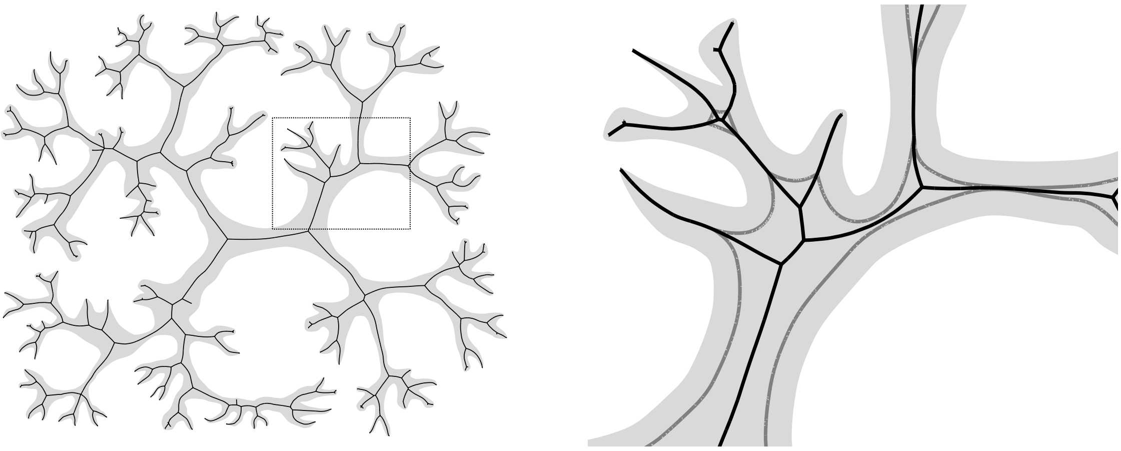
(b) Detail with offset

Figure 2.12: Austria shape

	double				**Gmpq**		
`# arcs`	`Divide`	`Conquer`	`total`	`Offset`	`Divide`	`Conquer`	`total`
838	0.14	0.02	0.15	0.02	9.01	0.42	9.43
1772	0.29	0.02	0.31	0.05	20.67	0.54	21.20
3564	0.63	0.03	0.66	0.09	45.97	0.93	46.90
7382	1.20	0.06	1.26	0.17	85.04	1.75	86.79
15614	2.50	0.13	2.64	0.35	183.31	3.33	186.63
33450	5.67	0.29	5.96	0.76	404.91	6.87	411.78
71876	10.88	0.57	11.45	1.61	867.09	14.25	881.34
154900	26.46	1.25	27.70	3.47	2011.97	29.24	2041.20
335122	57.85	2.61	60.46	7.49	3932.38	60.99	3993.37



The axis of the Austria shape depicted in Figure 2.12 is heavily branched and its boundary contains numerous non-smooth features. This bumpy boundary also induces a rather high average recursion depth of the decomposition step for the different approximations lying between 81.5 and 117.9. As can be seen in the log-log plot this results in a relative runtime that is worse than for other examples.

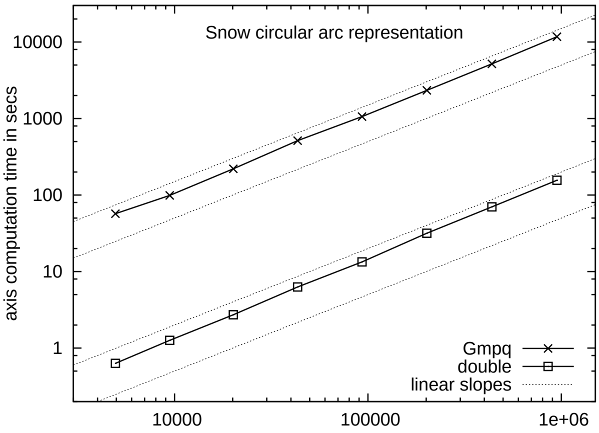


(a) Shape with axis

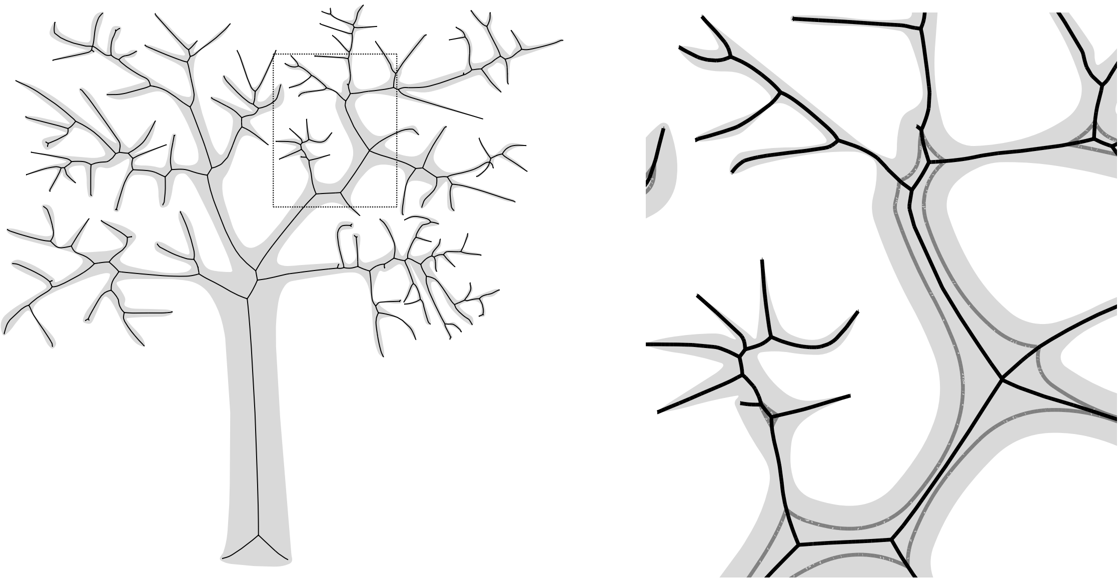
(b) Detail with offset

Figure 2.13: Snowflake shape

# arcs	double				Gmpq		
	Divide	Conquer	total	Offset	Divide	Conquer	total
4950	0.61	0.03	0.63	0.12	55.36	1.78	57.13
9440	1.21	0.05	1.26	0.21	96.25	2.62	98.87
20132	2.59	0.13	2.72	0.43	216.14	4.46	220.61
43332	6.01	0.29	6.29	0.89	504.19	8.71	512.90
93224	12.76	0.60	13.36	1.90	1039.95	17.78	1057.73
201688	30.45	1.27	31.71	4.05	2301.31	36.10	2337.41
437616	67.37	2.81	70.17	8.79	5094.80	75.65	5170.45
948706	149.93	5.91	155.84	18.98	11573.27	156.40	11729.67



The snow shape shown in Figure 2.13 has a medial axis similarly strongly branched as the one from Figure 2.12. However the smooth boundary of the snow shape induces a better performance of the algorithm and thus the resulting relative runtimes are also better than for the Austria shape as can be compared by the relative positions to the reference slopes in the according log-log plots.

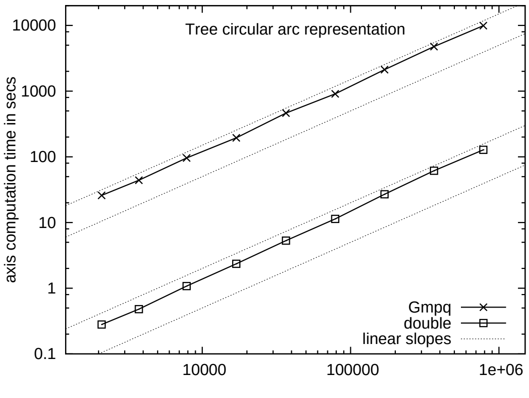


(a) Shape with axis

(b) Detail with offset

Figure 2.14: Tree shape

	double				**Gmpq**		
`# arcs`	`Divide`	`Conquer`	`total`	`Offset`	`Divide`	`Conquer`	`total`
2096	0.26	0.02	0.28	0.06	24.75	1.11	25.86
3736	0.44	0.04	0.48	0.09	42.68	1.38	44.06
7840	1.02	0.06	1.08	0.17	94.04	2.20	96.24
16970	2.24	0.10	2.35	0.35	190.29	3.95	194.24
36674	5.08	0.23	5.31	0.74	455.62	7.69	463.31
78736	10.88	0.48	11.36	1.58	894.97	15.37	910.34
169418	25.87	1.07	26.94	3.37	2091.90	31.95	2123.85
364528	59.08	2.28	61.36	7.25	4693.75	65.83	4759.58
784972	122.97	4.91	127.87	15.61	9772.96	138.96	9911.92



The structure and thus also the runtime behaviour of the medial axis algorithm for the tree shape is similar to the one of the snow shape from Figure 2.13. The average recursion depths of the decomposition step for the different approximations lie between 74.2 and 101.9. Considering that the combinatorially largest boundary approximation has over 370 times more arcs than the smallest one, this is a satisfactorily small fluctuation.

2.3 Voronoi diagram via medial axis

In this section we will extend the class of shapes for medial axis computation to domains with an arbitrary number of free-form holes (again represented by biarc boundaries).

Definition 2.3.1. *Given a connected shape* Ω *in the plane, the set* $\mathbb{R}^2 \setminus \Omega$ *consists of* ≥ 1 *connected components, whereas exactly one component is unbounded. A bounded component of* $\mathbb{R}^2 \setminus \Omega$ *is a simply connected closed set* h *which is henceforth denoted as a hole of* Ω*. In a set of holes* $\mathcal{H} = \{h_i\}$ *for a shape* Ω*, all* h_i *are pairwise disjoint topological disks of dimension two (shapes), one (curves) or zero (points).*

Let Γ be the outer part of the boundary of Ω which is not shared by a hole (the partial boundary of the unbounded connected component of $\mathbb{R}^2 \setminus \Omega$), then

$$\partial\Omega = \Gamma \cup \bigcup_i \partial h_i \ .$$

When considering the holes as Voronoi sites, the medial axis of the pierced shape can be used to derive the edge graph of a generalized Voronoi diagram, which is defined as follows.

The sites are the pairwise disjoint holes, being topological disks of dimension two, one, or zero in the Euclidean plane. That is, a site is either homeomorphic to a disk or to a line segment, or is simply a point. This includes polygons, circular disks, and open spline curves as sites. Here and throughout this section, let $\mathcal{S}$ denote the given set of sites, wheres $\mathcal{H} \subset \mathcal{S}$. The distance of a point x to a site $s \in \mathcal{S}$ is

$$d(x, s) = \min_{y \in s} \|x - y\| \ .$$

The Voronoi diagram, $V(\mathcal{S})$, of $\mathcal{S}$ is then defined via its *edge graph*, $\mathcal{G}_\mathcal{S}$, which is the set of all points having more than one closest point on the union of all sites. As in our case the sites are represented in a reasonable way (analytic curve pieces), this geometric graph is well defined [28]. An edge of $\mathcal{G}_\mathcal{S}$ containing points equidistant from two or more different points on the same site s is called a *self-edge* for s. The *regions* of $V(\mathcal{S})$ are the maximal connected subsets of the complement (of the closure) of $\mathcal{G}_\mathcal{S}$ in $\mathbb{R}^2$. They are topologically open sets.

Corollary 2. *The regions of* $V(\mathcal{S})$ *bijectively correspond to the sites in* $\mathcal{S}$*. Each site is contained in its region, and regions are simply connected.*

Proof. Let x be a point in the region R of $V(\mathcal{S})$. To x there exists a unique closest point, y, on the union of the sites in $\mathcal{S}$. (Otherwise, x would be a point on the edge graph $\mathcal{G}_\mathcal{S}$.) The sites are pairwise disjoint, so there is a unique site $s \in \mathcal{S}$ with $y \in s$. Site s is the same for all $x \in R$, because $d(x, s)$ is a continuous function of x. This maps regions to sites.

Now, obviously, with x also the closed line segment $\overline{xy}$ is part of R. This implies that R is simply connected. In particular, we have $y \in R$, which implies $s \subset R$ and maps sites to regions. □

We thus can talk of the region *of a site*, s, which will be denoted with $R(s)$ in the sequel. Henceforth the set of $\mathcal{S}$ will, beside the set of holes $\mathcal{H}$, also contain the surrounding boundary Γ of the pierced shape Ω. Then $V(\mathcal{S})$ can be derived from the medial axis of a pierced shape Ω as follows.

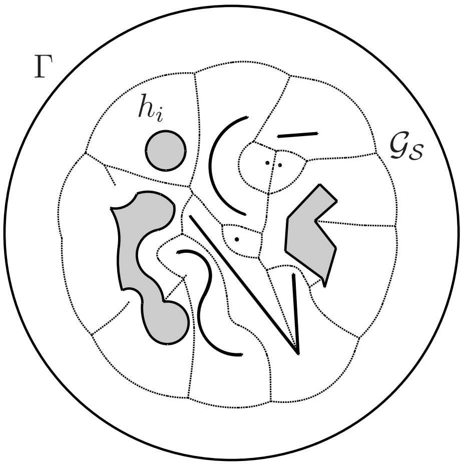


Figure 2.15: The edge graph $\mathcal{G}_\mathcal{S}$ of a Voronoi graph taking the holes $\{h_i\}$ and the outer boundary Γ of a shape as its sites. The sites are topological disks of dimension 0 (points), 1 (curves) and 2 (regions) with circular boundary representations. Note that the edge graph $\mathcal{G}_\mathcal{S}$ includes self-edges, described by points which have two closest points on the same site.

Observation 6. *Let* Ω *be a shape pierced by the set of holes* $\mathcal{H}$*, and with outer boundary* Γ*, and let* $\mathcal{S} = \mathcal{H} \cup \{\Gamma\}$ *be a set of Voronoi sites, then* $\mathrm{MA}(\Omega)$ *is equivalent to the closure of* $\mathcal{G}_\mathcal{S}$*.*[4]

All regions of $V(\mathcal{S})$ are bounded, except, of course, the region $R(\Gamma)$. Note that the outer curve Γ can always be chosen in a way such that each vertex of $V(\mathcal{S} \setminus \{\Gamma\})$ is also a vertex of $V(\mathcal{S})$.

The differences to a bisector-based definition of the Voronoi diagram should be noticed. Self-edges are ignored in such a definition unless the sites are split into suitable pieces. Such pieces, however, share boundaries—a fact that, if not treated with care, may give rise to unpleasant phenomena like two-dimensional bisectors.

2.3.1 Computation of medial axes with cycles

2.3.1.1 Where to break the cycles

The medial axis algorithm introduced in Section 2.2 works only on simply connected shapes, represented by one connected circular arc boundary. This is due to the fact that the medial axis of a shape Ω with holes can not be decomposed into two partial axes at every point of $\mathrm{MA}(\Omega)$, as it contains cycles.

For this reason we intend to show that removal of certain points on the medial axis $\mathrm{MA}(\Omega)$, being equivalent to $\mathcal{G}_\mathcal{S}$, breaks all its cycles. Finding such points is nontrivial, in view of the possible presence of self-edges. For a site $s \neq \Gamma$, let $p(s)$ be a point on s with smallest ordinate, and denote with $q(s)$ the closest point on $\mathcal{G}_\mathcal{S}$ vertically below $p(s)$. By the boundedness of $R(s)$, the point $q(s)$ always exists. Without loss of generality, it can be assumed that $q(s)$ is not an endpoint of any edge of $\mathcal{G}_\mathcal{S}$; this can always be achieved by rotating the coordinate system slightly.

Lemma 1. *For a set of sites* $\mathcal{S}$ *derived from a shape with holes as demonstrated in Observation 6, the geometric graph*

$$\mathcal{T}_\mathcal{S} = \mathcal{G}_\mathcal{S} \setminus \{q(s) \mid s \in \mathcal{S} \setminus \{\Gamma\}\}$$

---

[4] The reason why the closure of $\mathcal{G}_\mathcal{S}$ has to be considered lies in the possible existence of osculating maximal inscribed disks for Ω. The centers of such disks, while belonging to $\mathrm{MA}(\Omega)$, are not part of $\mathcal{G}_\mathcal{S}$. This subtle difference may be ignored for the purposes of this thesis.

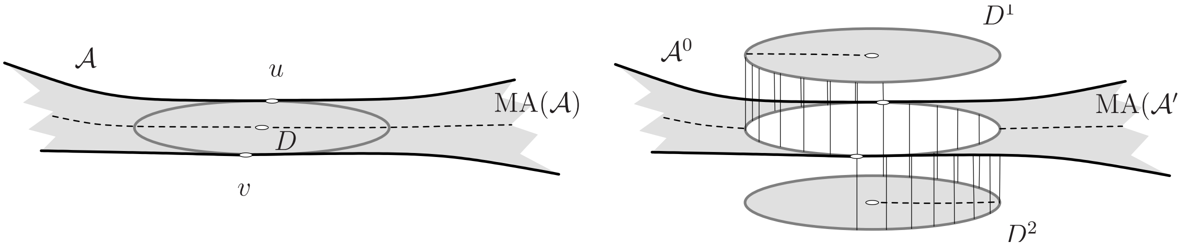


Figure 2.16: Domain $\mathcal{A}'$ (right) obtained as the augmentation of a given domain $\mathcal{A}$ (left) with a splitting maximal disk D. The medial axis (dashed) is split at the center of D.

is combinatorially a tree and thus cycle-free.

Proof. For each bounded region of $V(\mathcal{S})$, the edge graph $\mathcal{G}_\mathcal{S}$ contains a unique elementary cycle, because of the simple connectivity of regions (Corollary 2). For the same reason, the set of cycles does not change if self-edges are ignored. Interrupting each elementary cycle at a point vertically below its site leaves a geometric forest, because no path can continue below any site. Moreover, as these points are assumed to lie in the interiors of edges of $\mathcal{G}_\mathcal{S}$, each point destroys only one elementary cycle. That is, a geometric tree is obtained.

It remains to show that, for each site $s \neq \Gamma$, the point $q(s) \in \mathcal{G}_s$ does not lie on a self-edge for s. Recall that $q(s)$ is equidistant from $p(s)$ and from at least one other point, say y, on the union of all the sites. The ordinate of y is smaller than the ordinate of $p(s)$, because $p(s)$ lies vertically above $q(s)$. Thus, assuming that such a point y stems from s, which has to be the case if $q(s)$ lies on a self-edge for s, contradicts the definition of $p(s)$. □

2.3.1.2 Augmented domains

The goal is now to combinatorially disconnect the shape Ω at appropriate positions, such that the medial axis of the resulting domain corresponds to the tree decomposition $\mathcal{T}_\mathcal{S}$ of $V(\mathcal{S})$ as introduced in Lemma 1. As said (see [28]), a maximal inscribed disk can be used to split the medial axis of a simply connected shape into two components which share a point at the disk's center. In order to extend this result to shapes with holes, the notion of an *augmented domain* is introduced in a recursive manner as follows.

Definition 2.3.2. *An augmented domain is a set $\mathcal{A}$ together with a projection $\pi_\mathcal{A} : \mathcal{A} \to \mathbb{R}^2$. Initially, $\mathcal{A}$ is the original shape Ω, and the associated projection π_Ω is the identity. Now, consider a maximal disk D of an augmented domain $\mathcal{A}$, which touches the boundary $\partial\mathcal{A}$ in exactly two points u and v. Denote with $\widehat{uv}$ and $\widehat{vu}$ the two circular arcs which the boundary of D is split into. The new augmented shape, $\mathcal{A}'$, which is obtained from $\mathcal{A}$ by splitting it via the maximal disk D, is defined as*

$$\mathcal{A}' = \mathcal{A}^0 \cup D^1 \cup D^2$$

where $\mathcal{A}^0 = \{(x,0) \mid x \in A \setminus D\}$, $D^1 = \{(x,1) \mid x \in D\}$ and $D^2 = \{(x,2) \mid x \in D\}$. See Figure 2.16 for an illustration. The associated projection is

$$\pi_{\mathcal{A}'} : \mathcal{A}' \to \mathbb{R}^2,\ (x,i) \mapsto \pi_\mathcal{A}(x)\ .$$

We say that the line segment in $\mathcal{A}$ between points (x,i) and (y,j) is contained *in $\mathcal{A}'$ if one of the following conditions is satisfied:*

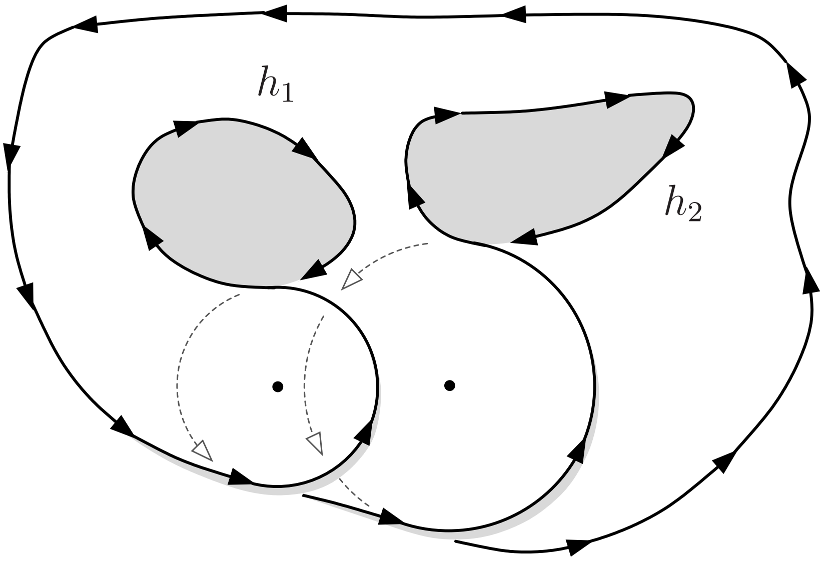


Figure 2.17: Oriented boundary of an augmented domain induced by two holes.

1. *$i = j$ and the line segment $\overline{xy}$ avoids ∂D,*
2. *$\{i, j\} = \{0, 1\}$ and $\overline{xy}$ intersects the arc $\overset{\frown}{uv}$, or*
3. *$\{i, j\} = \{0, 2\}$ and $\overline{xy}$ intersects the arc $\overset{\frown}{vu}$.*

For any two points (x, i) and (y, j) in $\mathcal{A}'$, their distance now can be defined. It equals the distance of $\pi_{\mathcal{A}}(x)$ and $\pi_{\mathcal{A}}(y)$ in $\mathbb{R}^2$, provided the connecting line segment is contained in $\mathcal{A}'$, and is ∞, otherwise. An (open) disk in $\mathcal{A}'$ with center (m, i) and radius ϱ is the set of all points in $\mathcal{A}'$ whose distance to (m, i) is less than ϱ. Such a disk is said to be *inscribed* in $\mathcal{A}'$ if its projection into $\mathbb{R}^2$ is again an open disk.

Having specified inscribed disks for $\mathcal{A}'$, the boundary of $\mathcal{A}'$ and the medial axis (transform) of $\mathcal{A}'$ can be defined as in the case of planar shapes. In particular, $\partial\mathcal{A}'$ derives from $\partial\mathcal{A}$ by disconnecting the latter boundary at the contact points u and v of the splitting disk D, and reconnecting it with the auxiliary circular arcs (see Definition 2.2.2) $\overset{\frown}{uv}$ and $\overset{\frown}{vu}$. This process is depicted schematically in Figure 2.17. Note that when $\partial\mathcal{A}'$ is traversed in a fixed orientation, the interior of $\mathcal{A}'$ stays on a fixed side.

Concerning the medial axis, every maximal inscribed disk in $\mathcal{A}$ different from D corresponds to exactly one maximal inscribed disk in $\mathcal{A}'$, hence there is a bijection between $\mathrm{MAT}(\mathcal{A}) \setminus \{D\}$ and $\mathrm{MAT}(\mathcal{A}') \setminus \{D^1, D^2\}$. The medial axis of $\mathcal{A}'$ therefore is the same geometric graph as $\mathrm{MA}(\mathcal{A})$, except that the edge of $\mathrm{MA}(\mathcal{A})$ containing the center of D is split into two disconnected edges which both have the center of D as one of their endpoints. These two points are two leaves of $\mathrm{MA}(\mathcal{A}')$; consult Figure 2.16 again.

To draw the connection to the edge graph $\mathcal{G}_{\mathcal{S}}$ of $V(\mathcal{S})$, the initial perforated shape Ω is augmented with $|\mathcal{S}| - 1$ maximal inscribed disks, namely, the ones centered at the points $q(s) \in \mathcal{G}_{\mathcal{S}}$ (for all $s \neq \Gamma$), where $q(s)$ was the vertical projection onto $\mathcal{G}_{\mathcal{S}}$ of a point with smallest ordinate on the site s. Denote with $\mathcal{A}_{\mathcal{S}}$ the resulting domain after these $|\mathcal{S}| - 1$ augmentation steps. The main finding of this section is concluded as follows.

Lemma 2. *The tree $\mathcal{T}_\mathcal{S}$ from Lemma 1 is the medial axis of the augmented domain $\mathcal{A}_\mathcal{S}$.*

2.3.1.3 How to break the cycles

Using Lemma 2, the Voronoi diagram $V(\mathcal{S})$ can be obtained by constructing the medial axis of the augmented domain $\mathcal{A}_\mathcal{S}$. The next aims are to show how to compute $\mathcal{A}_\mathcal{S}$ efficiently, and how to construct its medial axis without the need of calculating distances between points in $\mathcal{A}_\mathcal{S}$ directly.

Consider a planar perforated shape Ω whose augmentation has led to the augmented domain $\mathcal{A}_\mathcal{S}$. From the algorithmic point of view, augmenting Ω amounts to connecting its boundary $\partial\Omega$ to a *single* cyclic sequence, $\partial\mathcal{A}_\mathcal{S}$, that consists of pieces from $\partial\Omega$ and auxiliary arcs from circles bounding the splitting maximal disks. (One-dimensional sites contribute to $\partial\Omega$ with two curves, one for either orientation, and the special case of point sites can be handled consistently.) Each such boundary piece is used exactly once on $\partial\mathcal{A}_\mathcal{S}$, and traversing $\partial\mathcal{A}_\mathcal{S}$ corresponds to tracing the medial axis tree $\mathrm{MA}(\mathcal{A}_\mathcal{S})$ in preorder. See Figure 2.17, where a shape having two planar sites h_1 and h_2 as its holes is augmented with two disks, and the boundary of the resulting augmented domain is oriented for better visualization.

The construction of $\partial\mathcal{A}_\mathcal{S}$ is trivial once the splitting disks are available.[5] The main task is, therefore, to find these disks D_i, one for each site (hole in the shape) $h_i \in S \setminus \{\Gamma\}$. Recall from Section 2.3.1.1 that D_i is horizontally tangent to h_i at a lowest point $p(h_i)$ of h_i. The center $q(h_i)$ of D_i lies on the edge graph $\mathcal{G}_\mathcal{S}$ of $V(\mathcal{S})$ but, of course, D_i needs to be found without knowledge of $\mathcal{G}_\mathcal{S}$.

Indeed, a simple and efficient plane-sweep can be applied as follows. Sweep across $\mathcal{S}$ from above to below with a horizontal line L. For a site h_i, let x_i be the abscissa of $p(h_i)$, and define $E_L(i) = h_i \cap L$. Note that $E_L(i)$ may consist of more than one component. We maintain, for each site h_i whose point $p(h_i)$ has been swept over, the site h_j where $E_L(j)$ is closest to x_i on L. The unique disk with north pole $p(h_i)$ and touching h_j is computed, and the minimal such disk for h_i so far, $D_L(i)$, is updated if necessary. The abscissa x_i is deactivated again when $D_L(i)$ has been fully swept over by L.

Lemma 3. *After completion of the sweep, $D_L(i) = D_i$ holds for each site index i.*

Proof. For a fixed index i, let h_k be the site that defines the disk D_i. We have to show that $E_L(k)$ and x_i become neighbors on L while x_i is active. Consider a point t where D_i is tangent to h_k. Then, because D_i avoids all the sites, the line segment $\overline{x_i t} \subset D_i$ does the same. Thus $E_L(k)$ and x_i are adjacent when L passes through t. Also, x_i is active at this moment, $D_i \subset D_L(i)$ holds. $\square$

To keep small the number of neighbor pairs (x_i, h_j) on L processed during the plane sweep, only pairs where no other active abscissa x_m lies between x_i and $E_L(j)$ are considered; the disk $D_L(i)$ cannot have a contact beyond the one of $D_L(m)$, otherwise. The number of such pairs is linear. Thus the construction can be implemented in $\mathcal{O}(n \log n)$ time if the sites in $\mathcal{S}$ are described by a total of n objects, each being managable in constant time. Note that $\partial\mathcal{A}_\mathcal{S}$ then consists of $\Theta(n)$ pieces.

---

[5] As a possible degenerate case, a splitting disk may have more than two points of contact with the boundary $\partial\Omega$. In that case, any two contact points on different components of $\partial\Omega$ may be chosen. The algorithm that is described in the following automatically yields such a pair of points for each disk.

2.3.1.4 Computing the medial axis of $\mathcal{A}_{\mathcal{S}}$

Given the description of an augmented domain $\mathcal{A}'$ by its connected boundary (consisting of original and auxiliary arcs), it can be split into subdomains with the same property using maximal disks computed by `maxdisk`(a,$\partial\mathcal{A}'$) from Algorithm 1 for an arbitrary original arc a of $\partial\mathcal{A}'$.

Lemma 4. *The algorithm* `maxdisk`*(a,$\partial\mathcal{A}_{\mathcal{S}}$) correctly computes a dividing disk D for $\mathcal{A}_{\mathcal{S}}$ at an arbitrary point on the arc a.*

Proof. From Section 2.3.1.2 we know that the set of maximal inscribed disks is the same for $\mathcal{A}_{\mathcal{S}}$ and the perforated shape Ω, except for the (finitely many) disks D_i taking part in the augmentation. The assertion follows. □

In other words, the distances to the sites which are needed in the medial axis computation are the same in $\mathcal{A}_{\mathcal{S}}$ and in Ω. (This is, of course, not true for *all* possible distances.) Note that the auxiliary arcs are used only to link the site segments in the correct cyclic order; they do not play any geometric role (and do also not play a role for `maxdisk`).

The domain and its medial axis tree can again be split recursively, until the directly solvable base cases of the forms depicted in Figure 2.3 and Figure 2.4 remain. The decomposition of the augmented domain $\mathcal{A}_{\mathcal{S}}$ is evoked by calling `medax`($\mathcal{A}_{\mathcal{S}}$) described in Algorithm 2.

2.3.2 Practical Aspects

As argumented in Section 2.2.4, the algorithm `medax` performs best on medial axes with linear diameter and few branchings. In the case of Voronoi computation this corresponds to scenarios where a small number of sites is represented by a large number of individual objects (circular arcs in this case). The required accuracy for approximating the sites then typically leads to an input size that is independent from the branching of MA($\mathcal{A}_{\mathcal{S}}$). The tests in this context report small constants in the $\mathcal{O}(n \log n)$ term in this case. The implementation also shows favorable behaviour when compared to the demo package provided by CGAL for Voronoi computation with polygonal sites (see [69]). See Table 2.1 and Figure 2.18, where runtimes (averaged over 10 runs) for Voronoi computation of 40 sites with different input sizes are shown. The atomic step is the intersection test of a site-describing circular arc and a given disk, which is among the simplest imaginable tests when a closest-site Voronoi diagram is to be computed by means of distance calculations.

The other extreme is the case of many sites respectively represented by a small constant number of primitives. Here, by the way how $\mathcal{A}_{\mathcal{S}}$ is constructed, the diameter of MA($\mathcal{A}_{\mathcal{S}}$) will be typically much smaller. In this case the runtime development of the relevant CGAL demo program is (at least by trend) similar to the one of `medax`. In this worst case still a $\mathcal{O}(n\sqrt{n})$ behaviour of the algorithm can be guaranteed; see Table 2.2 and Figure 2.19.

`#arcs` (n)	medax($\mathcal{A}_\mathcal{S}$)			CGAL
	`seconds`	`atomic steps`	`ratio` $n \log n$	`seconds`
1714	0.19	45283	2.46	3.41
2824	0.31	72663	2.24	7.47
5622	0.66	166955	2.38	22.47
11822	1.40	355721	2.22	84.31
25210	3.16	819843	2.22	359.26
54214	7.00	1813780	2.13	1793.53
116460	16.08	4205066	2.15	11547.91
250366	35.62	9292356	2.07	126064.00
537360	80.73	21327696	2.09	> 1 week

Table 2.1: Computation times for the Voronoi diagram of 40 complex sites with different input sizes, the `medax` algorithm compared to an according CGAL demo package for polygons.

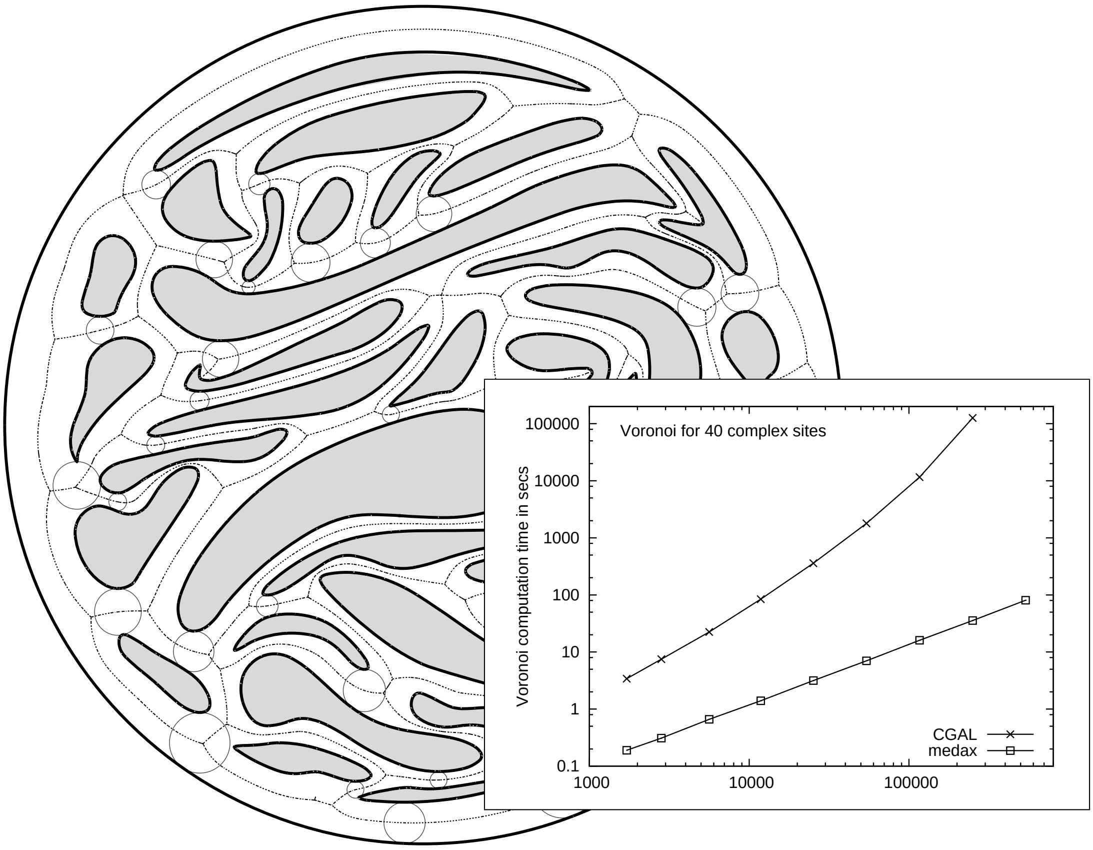


Figure 2.18: The medial axis of a shape with 40 holes. In grey the augmenting disks, which break all cycles of the medial axis. To the right the log-log plot graph to Table 2.1.

		`medax`$(\mathcal{A}_\mathcal{S})$			CGAL
`#sites`	`#arcs` (n)	`seconds`	`atomic steps`	`ratio` $n\sqrt{n}$	`seconds`
100	400	0.07	10036	1.25	1.12
500	2000	0.45	94163	1.05	6.02
1000	4000	0.99	228483	0.90	12.84
5000	20000	7.30	2017601	0.71	64.76
10000	40000	16.96	4891802	0.61	133.43
50000	200000	173.66	56629826	0.63	714.25
100000	400000	451.24	150649053	0.60	1429.26

Table 2.2: Computation times for the Voronoi diagram of very small sites with a constant number of arcs. The time needed by the sweep line algorithm for the computation of the augmenting disks is not included.

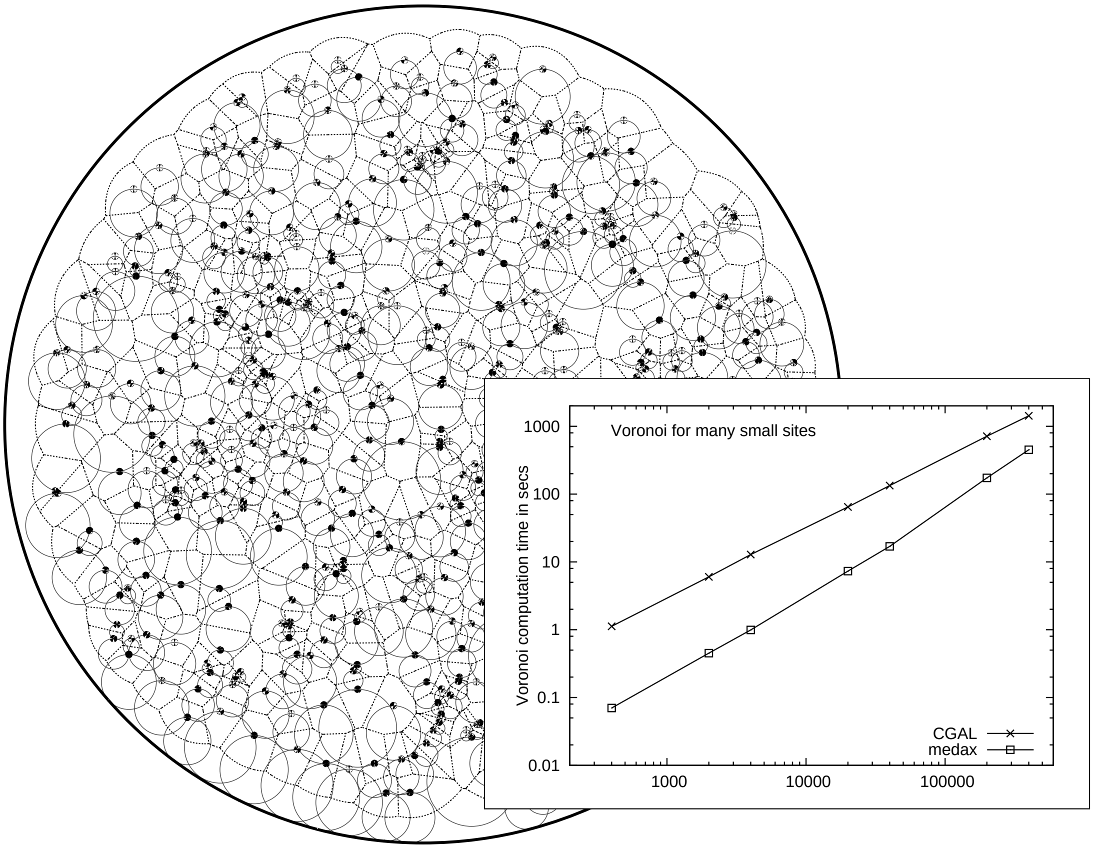
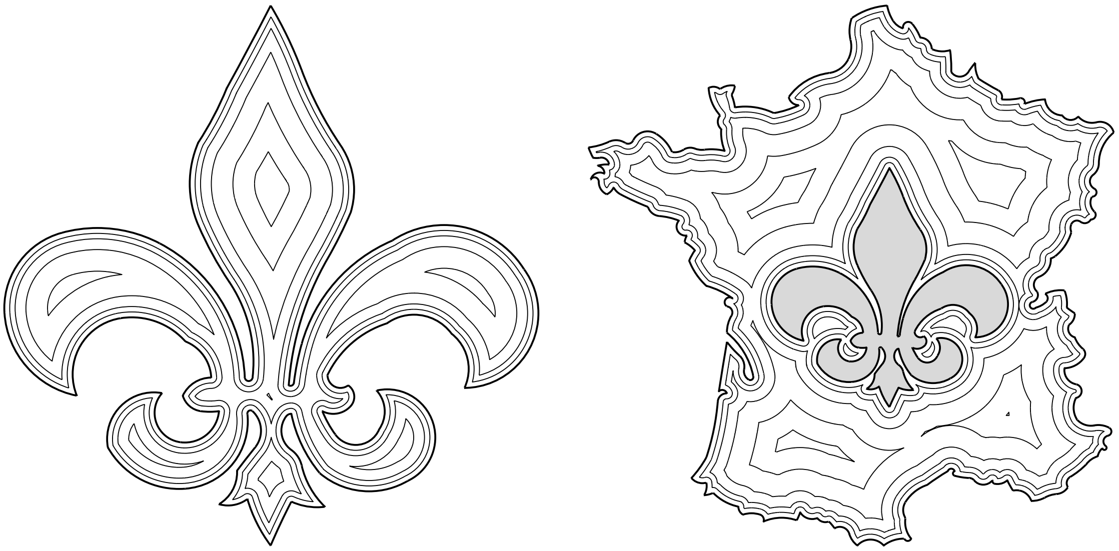


Figure 2.19: The main part of the Voronoi diagram for 500 small sites. In grey the augmenting disks, which break the diagram into a tree-like structure. To the right the log-log plot graph to Table 2.2.


(a) Inner offset

(b) Outer offset

Figure 2.20: To the left inner offsets of the planar *fleur-de-lis* shape for different values of ϱ as defined in Definition 1.2.7. If the same shape is embedded into another shape as a hole then also outer offsets can be obtained (to the right).

2.4 Trimmed offset computation

Offsetting is a fundamental operation for planar shapes, and it is needed frequently, e.g., in computer-aided manufacturing [63, 43]. Several authors base their offsetting algorithms on the Voronoi diagram or the medial axis [63, 32, 60, 23]. A piecewise-curvilinear representation of the input shape is advantageous, because the class of such shapes is closed under offsetting operations.

This means that for a shape Ω with circular boundary representation, the boundary $\partial\Omega^\varrho$ of an offset shape (see Definition 1.2.7) consists of circular arcs again, which are offsets of the circular arcs in $\partial\Omega$. However, simply offsetting $\partial\Omega$ does not give $\partial\Omega^\varrho$, since self-intersections may be present. The medial axis $\mathrm{MA}(\Omega)$ is used to trim away these self-intersecting parts.

2.4.1 The base case set-up

The subdivision into simple base cases (see Section 2.2.1), as resulting from the medial axis computation of simply connected shapes in Section 2.2 as well as from the construction of the Voronoi diagram of augmented domains in Section 2.3), is particularly well suited for the task of offset computation, because it delivers the necessary combinatorial structure without computing the edge graph explicitely.

As the structural information of the combinatorial medial axis is required for the trimming, only the inner offsets of a base case can be obtained. This allows inner offset computation for a given planar shape (see Figure 2.20a). For outer offsets, as shown in Figure 2.20b, a shape Ω can be defined as a hole within a suitable topological disk covering the shape, thus resulting in the inner offset of the complement of Ω.

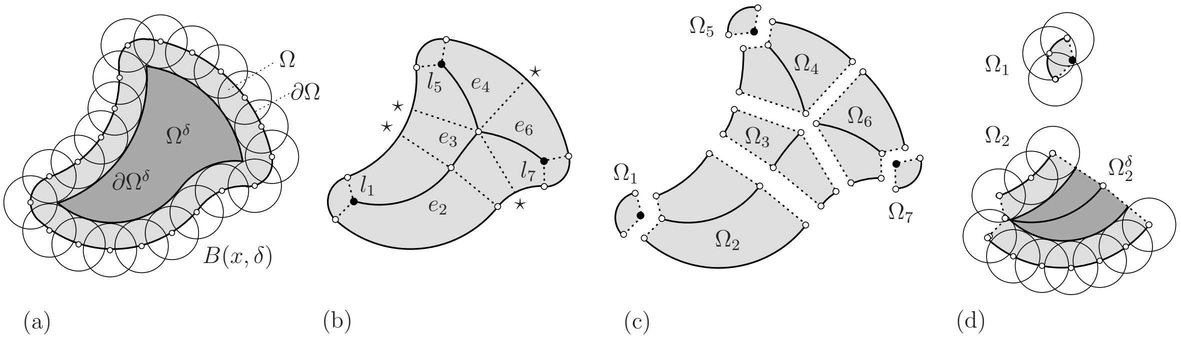


Figure 2.21: (a) Definition of the trimmed offset, (b) segmentation of the medial axis into edges and leaf points (additional arc endpoints are marked with $\star$), (c) segmentation of the shape, (d) offsets of subshapes.

Every edge of the medial axis corresponds (detected via the respective base case) to two defining primitives (point or arc) on the boundary of the original shape Ω (see Section 2.2.3). Every leaf point of the axis is assigned to a single arc on the boundary which defines a local curvature maximum (see Observation 5). The general idea is now, to find partial offset curves for every edge and leaf point of the medial axis, which, concatenated, give the complete trimmed offset.

2.4.2 Medial axis-induced subshapes

In Section 2 during the divide step, the idea was to find, via the boundary of the shape, a decomposition into simple base cases, which allow the computation of the medial axis. Now, I want to use the computed medial axis, to find a decomposition of the initial shape into subshapes which allows convenient offset computation.

The medial axis of a circular boundary representation consists of conic segments e_i. For every edge (or leaf) e_i of the medial axis $\mathrm{MA}(\Omega)$ we define a subshape Ω_i as follows.

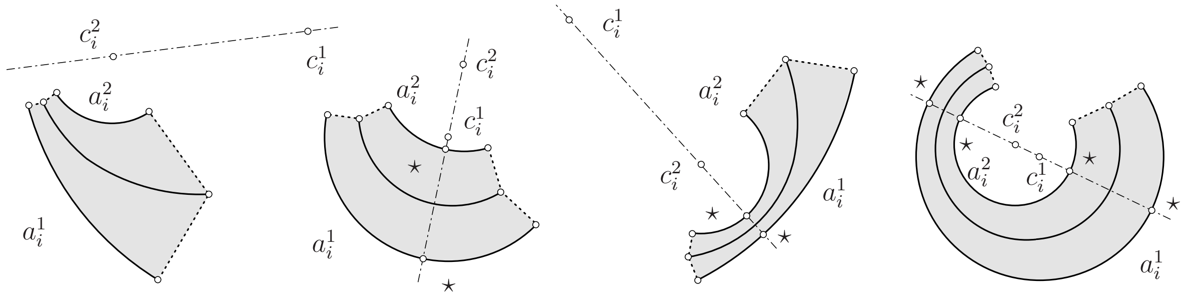
Definition 2.4.1. *Let e_i be an edge or a leaf point of the medial axis $\mathrm{MA}(\Omega)$. Let $\mathcal{D}_i$ be the set of all maximal disks D centered on e_i, and for a maximal disk D let $\mathcal{F}_D$ be the set of all footpoints of D on $\partial\Omega$. Then the subshape $\Omega_i \subseteq \Omega$,* induced by *$e_i$, is defined as the union over all segments, connecting the center c_D of a maximal disk with one of its footpoints, namely*

$$\Omega_i = \bigcup_{D \in \mathcal{D}_i} \left(\bigcup_{f \in \mathcal{F}_D} \overline{c_D f} \right) .$$

For an illustration of the above definition see Figure 2.21 (b) and (c).

Monotonic subshapes

A subshape Ω_i is said to be *monotonic* if the radii of the maximal disks of Ω with centers on e_i have no inner extrema. The extremal radii $r_{\min}, r_{\max}$ are then realized at the boundaries. Depending on the position (with respect to Ω_i) of the line L spanned by the centers of the



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

Figure 2.22: Splitting subshapes into monotonic pieces at the points marked with $\star$. The centers of the arcs a_i^1, a_i^2 span the dashed lines.

supporting circles of the arcs a_i^1, a_i^2, the radii have no, one, or two extrema.[6] The subshapes associated with leaves are already monotonic. Note that for splitting into monotonic subshapes we simply intersect a_i^1, a_i^2 with the line L, rather than computing the bisector of these arcs; see Figure 2.22.

2.4.3 Partial offset computation

The offsetting is done separately for each monotonic subshape. If $\varrho < r_{\min}$, then the offsets of the arcs at distance ϱ are fully contained in $\partial\Omega^\varrho$. For $r_{\min} \leq \varrho \leq r_{\max}$, the offset arcs are trimmed at their intersection; see Figure 2.21 (d), bottom. Finally, if $r_{\max} < \varrho$, then the subshape does not contribute to $\partial\Omega^\varrho$ (Figure 2.21 (d), top for a subshape defined by a leaf).

The implementation shows that offset computations require only little additional time after the medial axis construction; the tables for the examples Figure 2.10 to Figure 2.14 give several examples. The total time thus will not increase much in applications where many different offset layers are needed.

---

[6] If one of the arcs is a line segment, then L is the unequivocal line through the center of the supporting circle of the circular arc, perpendicular to the supporting line of the line segment. If both arcs are linear, then the subshape is always monotonic.

2.5 Exact computation for circular boundaries with rational coefficients

The Divide-&-Conquer approach presented in Section 2.2 has some minor drawbacks concerning exactness. This mainly results from the biarc construction of the boundary (as described in [2]), which provides theoretical smoothness, that is however not representable by usual float or rational number types. Further on, degenerate branching points of the medial axis cannot be detected exactly but have to be found with help of a proximity argument, as mentioned in Section 2.2.2.2.

In particular, the correct representation of the medial axis curve is a challenging task if the boundary input data does not comply with certain (numerical or algebraic) quality criteria as being rational representable or providing algebraically smooth joints between arcs.

An important part of most medial axis algorithms is the bisector computation. This problem has been approached for various types of rational curves, but mostly relying on machine arithmetic as in [41, 42].

In this section, the goal is to compute the algebraically correct medial axis. Thus, it has to be coped with exact bisector computation of (arc-supporting) circles. For this purpose we require all arcs on the boundary, that are involved in the bisector computation, to be *rational.* Arcs which do not directly contribute to the medial axis, but describe a local curvature maximum (see Observation 5) and thus merely a leaf-point of the axis, are allowed to be rational square-root expressions (*rasqex*).

Definition 2.5.1. *Integers are rasqex. If* x *and* y *are rasqex, so are* $x + y$, $x - y$, $x \cdot y$, x/y *and* $\sqrt{x}$.

The number type *rasqex* has exact comparison operators =, <, and >, realized in LEDA [22, 81] or the Core library [39, 40]. Actually, these two packages are able to represent arbitrary k-th root numbers, what is more than we need. For our purposes the *FieldWithSqrt* concept, which is part of the algebraic kernel provided by the CGAL library [64], is sufficient.

Several details of the algorithm, e.g. bisector and tritangent circle computation, are similar to those needed for the construction of an Apollonius diagram, as examined extensively in the work of Emiris and Karavelas [44]. They show that the operations allowed in the *rasqex* number type are sufficient to compute all predicates. Similar efforts have been made for ellipses and even more general smooth convex sites [45, 46].

Beside the similarities there are serveral additional aspects which have to be taken into account for the algebraically exact medial axis construction.

- The medial axis contains parts from the underlying bisectors different from the ones needed for the Apollonius diagram.
- In this work not an incremental approach as in [44] is pursued, but a Divide-&-Conquer algorithm is used. It is not obvious that all required steps therein can be accomplished with *rasqex* numbers.
- As opposed to the Apollonius diagram, medial axis computation does in general not deal with single sites and complete circles, but one closed curve (in this case composed of circular arcs) representing a planar shape.

Concerning the last point, we consider, for matters of clarity, only circular boundaries without line segments. The extension to boundaries with linear components would require an extension of the cases that may occur for bisector computation as well as for the Apollonius problem. To allow exact computation the boundary has to fulfill some additional criteria (discussed in Section 2.5.1), which qualifies it as a *Rational Circular Arc Boundary* or *RCAB*.

Given a *RCAB*, decomposition is done mainly as in the `medax` procedure (Algorithm 2), with only minor modifications of maximal disk construction, see Section 2.5.2.

For bisector computation in the conquer step, it is shown that these bisectors are algebraic curves of degree 4 over the rational numbers $\mathbb{Q}$, which can be expressed as the product of two quadratic polynomials (conics) over a simple extension field of $\mathbb{Q}$, see Section 2.5.2.

Finally in Section 2.5.4 the center points of auxiliary arcs (Section 2.5.2) and branching points (Section 2.5.4) are used to isolate relevant parts of the conic curves which contribute to the medial axis, completing the conquer-part of the algorithm.

2.5.1 Rational circular arc boundary

A circular boundary representation is defined in a form, that allows exact bisector and medial axis construction. This requires some definitions, starting with *rational circles.* Let $\mathbb{Q}$ denote the set of rational numbers.

Definition 2.5.2. *For a circle C with center c the following definitions are equivalent:*

$$\begin{aligned} C \textit{ is a rational circle} &\iff c \in \mathbb{Q}^2 \;\; \textit{and} \;\; \exists u \in C : u \in \mathbb{Q}^2 \\ &\iff \exists u, v, w \in \mathbb{Q}^2 : u, v, w \in C \; . \end{aligned}$$

Note that the squared radius of a rational circle is rational. It is also well-known that on a rational circle C points with rational coordinates are lying dense (see [82]). This means that near an arbitrary point p on C and for any $\varepsilon > 0$ one can find a rational point in an ε-environment around p, that lies on C. We say that an arc is *rational*, if its supporting circle and its two endpoints are rational. By extending to *rasqex* numbers, we can now define *rasqex circles* as a superset of rational circles.

Definition 2.5.3. *For a circle C with center c and squared radius r the following definitions are equivalent:*

$$C \textit{ is a rasqex circle} \iff c \textit{ and } r \textit{ are rasqex} \; .$$

An arc is *rasqex*, if its supporting circle and its two endpoints are *rasqex.* A rational circle is always a *rasqex* one, but not vice versa. For the C^1-boundary representation we want to rely on rational circles as much as possible. The demands are only softened by allowing *rasqex* arcs whenever they are not directly contributing to bisector calculation. This is true for arcs which describe a local curvature maximum, as such a maximum always defines a leaf-point of the medial axis. This means that such an arc does not contribute to any bisector computation later on, however its center point is eventually required for point location as shown in Section 2.5.4).

Definition 2.5.4. *Consider a circular arc boundary representation $\partial\Omega$. An arc that constitutes a local curvature maximum, and thus a leaf point of the medial axis, has to be at least rasqex. If all other arcs are rational, then we call $\partial\Omega$ a Rational Circular Arc Boundary (RCAB).*

The reason why these special arcs only have to be *rasqex* lies in the fact that the supporting circles of these arcs are not directly participating in bisector computation, but are only required for confining the relevant parts.

2.5.2 Exact construction of a maximal disk

The construction of a dividing maximal disk has to be done with care, to take advantage of the properties of rational arcs. The centers of the maximal disks (providing the auxiliary arcs) play an important role, as they are required to lie exactly on the bisector curves for segment confinement. A general maximal disk has two contact points on $\partial\Omega$, which lie on two different arcs of the *RCAB*.

We will now give details about the construction of a maximal disk with the required algebraic properties in comparison to the `maxdisk` procedure introduced in Section 2.2.1 and the details illustrated in Figure 2.2.

It is started by choosing a random arc a on the boundary, which is not allowed to be a local curvature maximum. As $\partial\Omega$ is a *RCAB*, a is rational and thus a rational point $\mathcal{P}$ as close to an arbitrary point on a as is seen fit can be chosen. See [21] for a detailed algorithm and implementation on how to choose such a point. For every arc $a_k \neq a$ of $\partial\Omega$ we construct the disk that touches a at $\mathcal{P}$ and is tangent to a_k (see Figure 2.2), as usual. The arc holding the second footpoint of the sought-after maximal disk D again has to be a rational one. The *RCAB* now guarantees certain algebraic properties of D. As the point c as well as $\mathcal{P}$ are rational, so is the line l. The radius r_k as shown in Figure 2.2 (being the square-root of the r from Definition 2.5.3) is not rational in general, as a consequence so isn't c'_k. However, $c'_k \in \mathbb{Q}(r_k)^2$. Therefore also the point of intersection between l and l', being the center of D, is in this extension field.

Corollary 3. *Consider a maximal disk D of a RCAB. Let D be tangent to the two arcs a_p and a_q, lying on the rational circles C_p and C_q with squared radii r_p and r_q respectively. If the footpoint of D on a_p has rational coordinates, then*

$$c_D \in \mathbb{Q}(\sqrt{r_q})^2 \ .$$

Values in $\mathbb{Q}(\sqrt{r_q})$ can be represented exactly by the *rasqex* numbers, which makes later point location on the bisector curves convenient (see Section 2.5.4). Furthermore, note that the (*rasqex*) auxiliary arcs, stemming from the (*rasqex*) boundary circle of a maximal dividing disks, always describe a local curvature maximum of a partial boundary $\partial\Omega_i$. This is coherent with Definition 2.5.4.

2.5.3 Exact bisector computation

We next show how to compute the bisector between two rational arcs in an algebraic way, avoiding any numerical errors.[7] It will be shown that the whole computation can be done over the field of rational numbers with only a few adjoint square-roots. Let C_p and C_q be the supporting circles of the two arcs with centers $c_p = (x_p, y_p)$ and $c_q = (x_q, y_q)$ and squared radii r_p and r_q, respectively:

---

[7] Note that a reflex point (if bisector-defining) can be regarded as a circle with radius zero for all computations in this section.

$$\begin{aligned} C_p(x,y) &:= (x-x_p)^2+(y-y_p)^2-r_p \\ C_q(x,y) &:= (x-x_q)^2+(y-y_q)^2-r_q \;. \end{aligned}$$

As before, we assume $c_p, c_q \in \mathbb{Q}^2$ and $r_p, r_q \in \mathbb{Q}$.

Definition 2.5.5. *The* bisector curve *between the two circles C_p and C_q consists of all points (x, y) in the plane for which*

$$\big||(x,y)-c_p| \pm \sqrt{r_p}\big| = \big||(x,y)-c_q| \pm \sqrt{r_q}\big|$$

Lemma 5. *The bisector curve of the two circles C_p and C_q factors into two curves $B_1(x, y) = 0$ and $B_2(x, y) = 0$ where*

$$\begin{aligned} B_1(x,y) &= (d_1^2+d_2^2-r^2)^2-4d_1^2d_2^2 \;\in\; \mathbb{Q}(\sqrt{r_pr_q})[x,y] \\ B_2(x,y) &= (d_1^2+d_2^2-\tilde{r}^2)^2-4d_1^2d_2^2 \;\in\; \mathbb{Q}(\sqrt{r_pr_q})[x,y] \end{aligned}$$

with $d_1 := d_1(x, y) := |(x, y) - c_p|$, $d_2 := d_2(x, y) := |(x, y) - c_q|$, $r := \sqrt{r_p} - \sqrt{r_q}$ and $\tilde{r} := \sqrt{r_p} + \sqrt{r_q}$.

Proof. For the bisector-curve there exist two cases:

$$\textit{Case 1}\;\left\{\begin{aligned} &\left\{\begin{array}{rcl} d_1+\sqrt{r_p} &=& d_2+\sqrt{r_q} \;\vee\; d_1-\sqrt{r_p} = d_2-\sqrt{r_q} \\ \vee\; d_1+\sqrt{r_p} &=& -d_2+\sqrt{r_q} \;\vee\; d_1-\sqrt{r_p} = -d_2-\sqrt{r_q} \end{array}\right. \\ \Longleftrightarrow\; &\left\{\begin{array}{rcl} d_1-d_2 &=& -r \;\vee\; d_1-d_2 = r \\ \vee\; d_1+d_2 &=& -r \;\vee\; d_1+d_2 = r \end{array}\right. \quad |\_^2 \\ \Longleftrightarrow\; & d_1^2+d_2^2-r^2 = 2d_1d_2 \;\vee\; d_1^2+d_2^2-r^2 = -\,2d_1d_2 \quad |\_^2 \\ \Longleftrightarrow\; & (d_1^2+d_2^2-r^2)^2 = 4d_1^2d_2^2 \end{aligned}\right.$$

This is exactly the equation for $B_1(x, y) = 0$. Similar for $B_2(x, y) = 0$:

$$\textit{Case 2}\;\left\{\begin{aligned} &\left\{\begin{array}{rcl} d_1+\sqrt{r_p} &=& -d_2-\sqrt{r_q} \;\vee\; d_1-\sqrt{r_p} = -d_2+\sqrt{r_q} \\ \vee\; d_1+\sqrt{r_p} &=& d_2-\sqrt{r_q} \;\vee\; d_1-\sqrt{r_p} = d_2+\sqrt{r_q} \end{array}\right. \\ \Longleftrightarrow\; & (d_1^2+d_2^2-\tilde{r}^2)^2 = 4d_1^2d_2^2 \end{aligned}\right.$$

Since $d_1^2, d_2^2 \in \mathbb{Q}[x, y]$, $\tilde{r}^2 = (\sqrt{r_p} + \sqrt{r_q})^2 = r_p + 2\sqrt{r_pr_q} + r_q \in \mathbb{Q}(\sqrt{r_pr_q})$, and $r^2 = (\sqrt{r_p} - \sqrt{r_q})^2 = r_p - 2\sqrt{r_pr_q} + r_q \in \mathbb{Q}(\sqrt{r_pr_q})$, it follows that $B_1(x, y), B_2(x, y) \in \mathbb{Q}(\sqrt{r_pr_q})[x, y]$. □

From now on B_1 and B_2 denote the curves described by $B_1(x, y) = 0$ and $B_2(x, y) = 0$ respectively. This allows us to prove the following well-known fact in a convenient way:

Lemma 6. *B_1 and B_2 in Lemma 5 are conics, i.e. planar curves of degree two.*

Proof. It suffices to prove that B_1 and B_2 are conics when the centers of the circles C_p and C_q are x-axis aligned and lie symmetrically on both sides of the y-axis:

$$C_p(x,y) := (x+d)^2+y^2-r_p, \quad C_q(x,y) := (x-d)^2+y^2-r_q \;.$$

This is no restriction because every pair of circles with d being half the distance between their two center points can be moved to this position by rotation and translation. B_1 and B_2 are then subject to the same transformation which does not change their degrees.

For C_p and C_q being in this special position it is

$$\begin{aligned} d_1^2 &= d_1^2(x,y) = \mid (x,y) - c_p \mid^2 = \mid (x,y) - (-d,0) \mid^2 = (x+d)^2 + y^2 \\ d_2^2 &= d_2^2(x,y) = \mid (x,y) - c_q \mid^2 = \mid (x,y) - (d,0) \mid^2 = (x-d)^2 + y^2 \ . \end{aligned}$$

This yields for the two cases

$$\textit{Case 1} \quad \left\{ \begin{array}{rl} & (d_1^2 + d_2^2 - r^2)^2 = 4d_1^2 d_2^2 \\ \Longleftrightarrow & (x^2 + d^2 + y^2 - \frac{r^2}{2})^2 = ((x+d)^2 + y^2)((x-d)^2 + y^2) \\ \Longleftrightarrow & 0 = 4x^2d^2 - x^2r^2 - d^2r^2 - y^2r^2 + \frac{r^4}{4} \end{array} \right.$$

This is the quadratic equation for B_1.

$$\textit{Case 2} \quad \left\{ \begin{array}{rl} & (d_1^2 + d_2^2 - \tilde{r}^2)^2 = 4d_1^2 d_2^2 \\ \Longleftrightarrow & 0 = 4x^2d^2 - x^2\tilde{r}^2 - d^2\tilde{r}^2 - y^2\tilde{r}^2 + \frac{\tilde{r}^4}{4} \end{array} \right.$$

This is the quadratic equation for B_2. □

Altogether Lemma 5 and Lemma 6 lead up to the main result:

Theorem 1. *The bisector of the two circles C_p and C_q in our original coordinate system factors into two conics over the field $\mathbb{Q}(\sqrt{r_p r_q})$ which is in rasqex.*

Corollary 4. *Each of B_1 and B_2 is either a hyperbola or an ellipse or a pair of identical lines.*[8]

Proof. Looking further at the equations for B_1 and B_2 in the special case where the center-points are x-axis aligned we first observe that B_1 and B_2 are the two conics

$$B_1(x,y) := bx^2 - ay^2 - ab, \qquad B_2(x,y) := \tilde{b}x^2 - \tilde{a}y^2 - \tilde{a}\tilde{b}$$

with

$$a = \frac{(\sqrt{r_p} - \sqrt{r_q})^2}{4} = \frac{r^2}{4}, \qquad b = d^2 - a = d^2 - \frac{r^2}{4}$$

and

$$\tilde{a} = \frac{(\sqrt{r_p} + \sqrt{r_q})^2}{4} = \frac{\tilde{r}^2}{4}, \qquad \tilde{b} = d^2 - \tilde{a} = d^2 - \frac{\tilde{r}^2}{4} \ .$$

First consider B_1. If $r_p = r_q$ we have $a = 0$, $b = d^2$ and $B_1(x,y) = d^2x^2$ consists of two identical lines along the y-axis. If $r_p \neq r_q$ it is true that $a > 0$ and

$$b > 0 \ \Leftrightarrow \ d^2 > \frac{r^2}{4} \ \Leftrightarrow \ 4d^2 > (\sqrt{r_p} - \sqrt{r_q})^2 \ \Leftrightarrow \ 2d > |\sqrt{r_p} - \sqrt{r_q}| \ .$$

That means,

---

[8] For concentrical circles C_p and C_q the ellipse degenerates to a circle. For circles with infinite radius (to represent e.g. linear segments) also parabolas would occur.

- if $2d > |\sqrt{r_p} - \sqrt{r_q}|$, then $b > 0$ and B_1 is an hyperbola,
- if $2d = |\sqrt{r_p} - \sqrt{r_q}|$, then $b = 0$ and $B_1(x, y) = -ay^2$ consists of two identical lines along the x-axis,
- if $2d < |\sqrt{r_p} - \sqrt{r_q}|$, then $b < 0$ and B_1 is an ellipse.

For B_2 we always have $\tilde{a} > 0$ and

$$\tilde{b} > 0 \;\Leftrightarrow\; d^2 > \frac{\tilde{r}^2}{4} \;\Leftrightarrow\; 4d^2 > (\sqrt{r_p} + \sqrt{r_q})^2 \;\Leftrightarrow\; 2d > \sqrt{r_p} + \sqrt{r_q}\ .$$

- The two circles C_p and C_q do not intersect iff $2d > \sqrt{r_p} + \sqrt{r_q}$. In this case $\tilde{b} > 0$ and B_2 is a hyperbola.
- C_p and C_q touch tangentially iff $2d = \sqrt{r_p} + \sqrt{r_q}$. Then $B_2(x, y) = -\tilde{a}y^2$ consists of two identical lines along the x-axis.
- C_p and C_q intersect iff $2d < \sqrt{r_p} + \sqrt{r_q}$. In this case $\tilde{b} < 0$ and B_2 is an ellipse.

□

2.5.4 Partial axis confinement

In order to compute and represent the medial axis of the exact circular arc boundary it must be possible to analyze a bisector-conic over the extension field $\mathbb{Q}(\sqrt{r_p r_q})$. This means that in a so called *one-curve analysis* a bisector-conic B, described by $B(x, y) \in \mathbb{Q}(\sqrt{r_p r_q})[x, y]$, is divided into x-monotone arcs. This is not difficult and works analogously to the one-curve analysis of a conic over $\mathbb{Q}$ described in [16]. The bisector-conic B is split at its x-extreme points, that are points where $B(x, y)$ and the partial derivative $B(x, y)_y = \frac{\partial B(x,y)}{\partial y}$ vanish simultaneously. If the bisector-conic consists of a pair of identical lines, we make the defining polynomial square-free. Now every resulting x-monotone arc can be represented by a tuple $([le, ri], nr)$, where le and ri are the x-coordinates of the left and right endpoint, respectively. Since le and ri are roots of quadratic polynomials over $\mathbb{Q}(\sqrt{r_p r_q})[x]$, they can be represented by *rasqex* numbers. The branch number nr is either 0 or 1 and indicates which of the two x-monotone arcs of the curve above the x-interval $[le, ri]$ is meant.

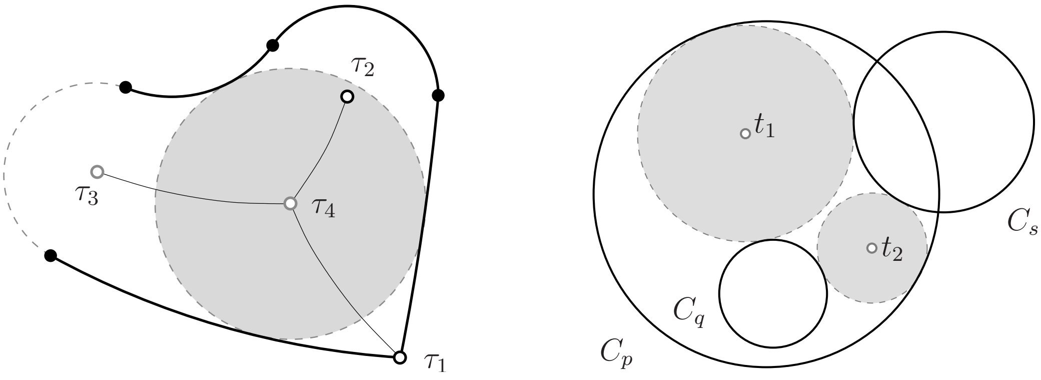
As mentioned in Section 2.5.2, point-location is an important issue. For a given point $t = (t_x, t_y)$, the coordinates of which are *rasqex*, the x-monotone arc of B_1 or B_2 it lies on has to be determined. First of all it is checked whether t lies on B_1 or B_2 by testing

$$B_1(t_x, t_y) = 0 \quad \text{or} \quad B_2(t_x, t_y) = 0\ . \tag{2.1}$$

Since all the numbers in $B_1(t_x, t_y)$ and $B_2(t_x, t_y)$ are *rasqex* numbers, the exact test for zero can be realized by using the equality operator of the *rasqex* numbers. Assume that p lies on B_1. The $<$-operator of the *rasqex* numbers is used to determine the two x-monotone arcs of B_1 for which

$$le \leq t_x \leq ri\ . \tag{2.2}$$

The last step is to determine whether t lies on the upper or lower branch, i.e., algebraically whether t_y is the greater or smaller root of the polynomial $B_1(t_x, y)$. Since $B_1(t_x, y)$ is a



(a) Limiting points

(b) Tritangent disks

Figure 2.23: (a) Different types of points used for conic curve confinement, (b) two disks with center t_1 and t_2 tritangent to C_p, C_q and C_s.

quadratic polynomial the coefficients of which are *rasqex*, its two roots r_1 and r_2 can be computed symbolically by introducing a new square-root. Now it has to be checked whether

$$t_y - r_1 = 0 \quad \text{or} \quad t_y - r_2 = 0 \ . \tag{2.3}$$

Again this can be done by using *rasqex* numbers. Notice that in cases where locally around t neither the second bisector-conic nor the second arc pass by and all x-extreme points are far away, the three steps for point-location can be sped up by using isolating intervals for t_x and t_y and evaluating the expressions in (2.1), (2.2) and (2.3) with interval arithmetic, if desired.

The medial axis of a circular boundary representation is computed as the union of bisector-conic segments. Each conic segment is limited by center points of maximal disks. Maximal disks of interest in this context are of four different forms, as illustrated in Figure 2.23a.

τ_1 : A convex point on $\partial\Omega$ (inducing a maximal disk with radius zero) confines a conic curve of the axis. Such a point is always a joint between two arcs, and thus *rasqex*.

τ_2 : The bounding circle of a limiting maximal disk may support an arc on the boundary which defines a local curvature maximum. From Definition 2.5.4 it can be assumed that such a point is *rasqex*.

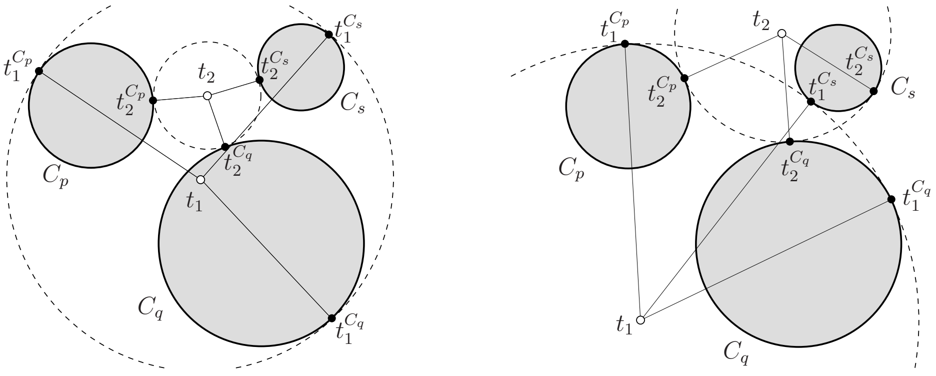
τ_3 : Center points of maximal disks which induce auxiliary arcs during the decomposition process are needed to confine axis curves. In Section 2.5.2 it is guaranteed that these points are *rasqex*.

τ_4 : Center points of tritangent disks are used for point location and subsequent confinement as well.

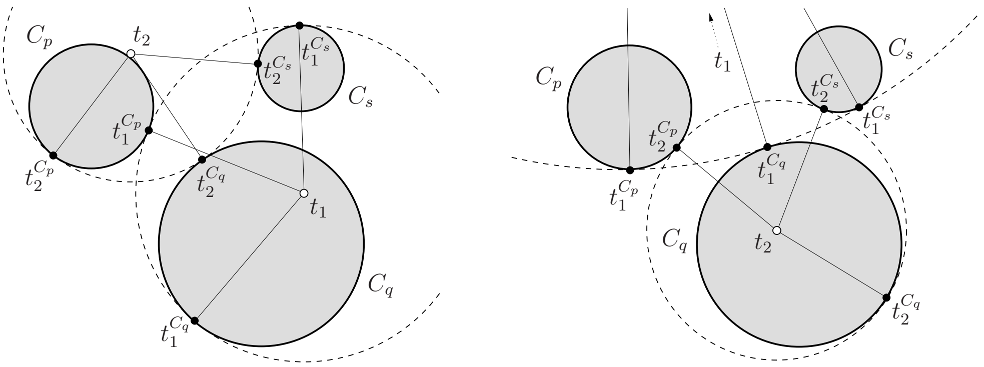
In the following section it is shown, that also center points of maximal disks with more than two points of tangency have *rasqex* coordinates, if $\partial\Omega$ is a *RCAB*.

Apollonius: tritangent circles

As said in the above section, points of type τ_4, being the branching points of the medial axis, are needed for the confinement of axis curves. The intention is to show that these points are *rasqex*, if the supporting circles of the defining arcs are rational.



(a) $\Phi(t_i) \in \{(0,0,0),(1,1,1)\}$ (b) $\Phi(t_i) \in \{(0,0,1),(1,0,0)\}$



(c) $\Phi(t_i) \in \{(1,0,0),(0,1,1)\}$ (d) $\Phi(t_i) \in \{(1,0,1),(0,1,0)\}$

Figure 2.24: Three fixed circles C_p, C_q and C_s with all eight possible circles, that are tangent to the former three. If the segment defined by the two points t_i and $t_i^{C_j}$ overlaps the disk bounded by the circle C_j then the tritangent disk with center t_i is called *inner tangent* to C_j, otherwise *outer tangent*.

It suffices to consider tritangent maximal disks, as any branching point of higher valency is defined by at least three rational arcs. A bisector curve between two rational circles is an algebraic curve of degree 4, and the branching point is one of the intersection points where all three bisectors between three circles meet.

There are two variations how a point on a bisector-curve may describe tangency at its footpoint on a defining circle.

Definition 2.5.6. *Consider a bisector-curve B and one of its two defining circles C. For a point $t \in B$ let t^C be its unequivocal footpoint on C and Γ_C the open region bounded by C. Then the function $\varphi(t, C)$ is defined on B as follows:*

$$\varphi(t,C) = \begin{cases} 0 & \textit{if} \quad \overline{t\ t^C} \cap \Gamma_C = \emptyset \\ 1 & \textit{otherwise} \end{cases}$$

If $\varphi(t, C)$ is 0 then the circle with center t and radius $\overline{t\ t^C}$ is called outer tangent *to C, otherwise* inner tangent.

As proved in Section 2.5.3 every bisector B consists of two conic curves, B_1 and B_2. By construction, the points on these two conics have certain properties concerning the function φ which are investigated next.

Lemma 7. *Consider the bisector B consisting of the two bisector-conics B_1 and B_2 and its two defining circles C_p and C_q, then*

$$\begin{aligned} \forall t \in B_1 \quad &: \quad (\varphi(t, C_p), \varphi(t, C_q)) \in \{(0,0), (1,1)\} \\ \forall t \in B_2 \quad &: \quad (\varphi(t, C_p), \varphi(t, C_q)) \in \{(0,1), (1,0)\} \ . \end{aligned}$$

Proof. As derived in the proof of Lemma 5, for every point t on B_1 it holds that

$$\begin{aligned} &|t - c_p| + \sqrt{r_p} = |t - c_q| + \sqrt{r_q} \quad \vee \quad |t - c_p| - \sqrt{r_p} = |t - c_q| - \sqrt{r_q} \\ \vee \quad &|t - c_p| + \sqrt{r_p} = -|t - c_q| + \sqrt{r_q} \quad \vee \quad |t - c_p| - \sqrt{r_p} = -|t - c_q| - \sqrt{r_q} \ . \end{aligned}$$

This leads to

$$\begin{aligned} &\varphi(t, C_p) = 1 \wedge \varphi(t, C_q) = 1 \quad \vee \quad \varphi(t, C_p) = 0 \wedge \varphi(t, C_q) = 0 \\ \vee \quad &\varphi(t, C_p) = 1 \wedge \varphi(t, C_q) = 1 \quad \vee \quad \varphi(t, C_p) = 1 \wedge \varphi(t, C_q) = 1 \ . \end{aligned}$$

For every point t on B_2 it is

$$\begin{aligned} &|t - c_p| + \sqrt{r_p} = -|t - c_q| - \sqrt{r_q} \quad \vee \quad |t - c_p| - \sqrt{r_p} = -|t - c_q| + \sqrt{r_q} \\ \vee \quad &|t - c_p| + \sqrt{r_p} = |t - c_q| - \sqrt{r_q} \quad \vee \quad |t - c_p| - \sqrt{r_p} = |t - c_q| + \sqrt{r_q} \ . \end{aligned}$$

This leads to

$$\begin{aligned} &\text{undefined} \quad \vee \quad \varphi(t, C_p) = 0/1 \wedge \varphi(t, C_q) = 1/0 \\ \vee \quad &\varphi(t, C_p) = 1 \wedge \varphi(t, C_q) = 0 \quad \vee \quad \varphi(t, C_p) = 0 \wedge \varphi(t, C_q) = 1 \ . \end{aligned}$$

□

Of interest is the situation where three rational circles C_p, C_q and C_s are given. They define three bisectors: $B^{'}$ between C_p and C_q, $B^{''}$ between C_q and C_s and $B^{'''}$ between C_p and C_s. A branching point t^*, being the center of a tritangent circle, lies on all three bisectors and so $\varphi(t^*, C)$ is well defined for $C \in C_p, C_q, C_s$. Let

$$\Phi(t^*) := (\varphi(t^*, C_p), \varphi(t^*, C_q), \varphi(t^*, C_s)) \ . \tag{2.4}$$

Observation 7. *Depending on which bisector-conics intersect in a branching point t^*, it can be distinguished between four different sets of contact tuples (see Figure 2.24 for illustration). For all other possible combinations of three bisector-conics a common intersection point is impossible.*

$$t^* \in B_1^{'} \cap B_1^{''} \cap B_1^{'''} \quad \Rightarrow \quad \Phi(t^*) \in \{(0,0,0), (1,1,1)\} \tag{2.5}$$

$$t^* \in B_1^{'} \cap B_2^{''} \cap B_2^{'''} \quad \Rightarrow \quad \Phi(t^*) \in \{(0,0,1), (1,1,0)\} \tag{2.6}$$

$$t^* \in B_2^{'} \cap B_1^{''} \cap B_2^{'''} \quad \Rightarrow \quad \Phi(t^*) \in \{(1,0,0), (0,1,1)\} \tag{2.7}$$

$$t^* \in B_2^{'} \cap B_2^{''} \cap B_1^{'''} \quad \Rightarrow \quad \Phi(t^*) \in \{(1,0,1), (0,1,0)\} \tag{2.8}$$

For example, considering case (2.5), if $t^* \in B_1^{'} \cap B_1^{''} \cap B_1^{'''}$, then due to Lemma 7 it holds that

$$\begin{aligned} & (\varphi(t^*, C_p), \varphi(t^*, C_q)) \in \{(0,0),(1,1)\} \\ \wedge \quad & (\varphi(t^*, C_q), \varphi(t^*, C_s)) \in \{(0,0),(1,1)\} \\ \wedge \quad & (\varphi(t^*, C_p), \varphi(t^*, C_s)) \in \{(0,0),(1,1)\} \ . \end{aligned}$$

This is only true if $\varphi(t^*, C_p) = \varphi(t^*, C_q) = \varphi(t^*, C_s) = 0$ or $\varphi(t^*, C_p) = \varphi(t^*, C_q) = \varphi(t^*, C_s) = 1$. The other cases work analogously.

The construction of all possible circles that are tangent to three given circles is a much discussed topic, with various possible ways of solution (see e.g. [75]). There exist at most 8 different circles that are tangent to three circles in the plane (see Figure 2.24).[9] The Gergonne construction, named after french mathematician Joseph Diaz Gergonne, is based on inverse geometry and uses so-called *lines of similitude.* For three circles in general position, there exist 4 lines of similitude. Each line induces at most 2 tritangent circles, which can both together be assigned to one specific case (2.5) to (2.8) from Observation 7. Note however, that e.g. for case (2.7) there may be two solutions of the form $(1,0,0)$ and none for $(0,1,1)$ (see Figure 2.23b for an example).

Altogether this means that constellations of three bisector-conics as shown in Observation 7 have at most two common intersection points. The x-coordinates of the intersection points of two of the three conics are roots of a degree four polynomial P_1 (which can be derived by a resultant computation). For another pair of conics we obtain another polynomial P_2. We now isolate the common x-components by computing the greatest common divisor $P' = \gcd(P_1, P_2)$. As at most two common solutions may exist, P' is a quadratic polynomial. Its roots can be represented exactly by *rasqex* numbers.[10] The same way the possibly two y-coordinates can be computed. This shows that the coordinates of the center points of tritangent circles can be represented as *rasqex* numbers and we get $2 \cdot 2 = 4$ candidates for them.

2.5.5 Handling degeneracies

In the general approach a maximal disk that is tangent to more than three boundary arcs cannot be detected accurately, but is instantiated so to speak on spec if a ε-proximity criterion is satisfied (Section 2.2.2.2).

Granted algebraic correctness such degenerate cases can be detected more easily. When arriving during decomposition at a partial shape, whose boundary is an alternating sequence of auxiliary and original arcs, then all bisectors between original arcs that are only separated by a single auxiliary arc are computed. If all these bisectors intersect in one single point then a degenerate case has occured. Computation is based on the principle introduced in Section 2.5.4, meaning that again *rasqex* numbers are sufficient for exact calculation. This elegant and intuitive handling of branching points with valency > 3 is one of the main advantages of the algebraically correct approach to medial axis computation.

---

[9] The classical Apollonius problem deals with three sites in a more general form, namely points, lines and circles. Points can be interpreted as circles with zero radius, for lines more case studies are necessary.

[10] In the special case where P_1 and P_2 have more than two common roots due to covertical intersection points, we shear the coordinate system, compute the center points of the tritangent circles in the sheared system and transform the result back to the original coordinate system.

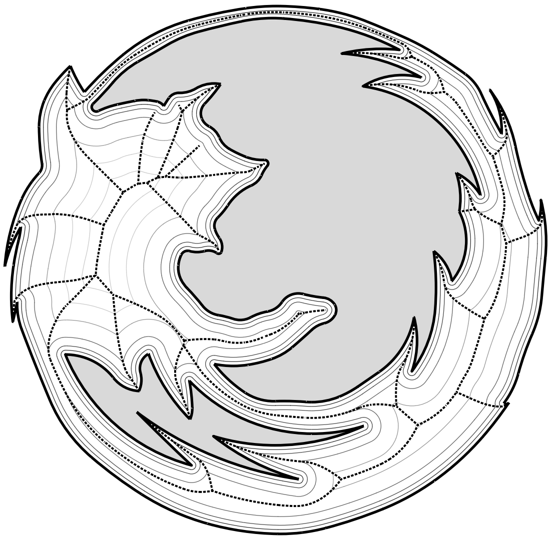


Figure 2.25: The axis and offsets of this figure, as of all other examples up to this point, that depict complete shapes have been computed with the algorithms described in the previous sections.

The Firefox logo® is a registered trademark of Mozilla® [88].

2.6 Conclusion of Chapter 2

In Section 2.2 we provide the theory and construction details for an efficient and stable implementation of a medial axis algorithm for planar, simply connected free-form shapes. To our knowledge this is the first implementation that runs fast in practice and, combined with a biarc approximation of increasing accuracy, converges to the exact medial axis of the input shape. The program can compete with current state-of-the-art implementations on this field with regard to correctness, speed, and reliability. The basic idea was using a piecewise circular boundary conversion, which allows for appropriate feature preservation of the shape, as well as for a simple and fast medial axis algorithm. Expected runtime for a shape with increasing input value n of boundary arcs is $\mathcal{O}(n \log n)$, a worst-case runtime of $n\sqrt{n}$ can always be guaranteed.

In Section 2.3 this algorithm is extended for shapes with arbitrary holes, which can be adapted for Voronoi computation of very general sites which are allowed to be topological disks of dimension zero, one and two. While classical Voronoi algorithms are in general fast on small sites, the merits of Voronoi computation via a medial axis construction reveal themselves when dealing with fewer sites, represented by a high number of boundary primitives. The performance is favorable when compared to other algorithms (e.g. the relevant demo CGAL package [69]) in this context.

The applicability of a circular boundary as well as the representation of the shape and its medial axis via simple base cases for trimmed offset computation is explored in Section 2.4. Not only is the described approach very intuitive and simple, but also performs, given that the medial axis is already computed, very fast (linear in the number of edges of the axis).

Finally, in Section 2.5, the algebraically exact computation of the medial axis is explained for boundaries essentially composed of rational arcs (*RCAB*). It is shown that all necessary computations can be performed over the field of rational numbers with a small number of adjoint square-roots. The defined *rasqex* number type can be realized in existing libraries as LEDA [81] and the Core library [39] (supported by e.g. CGAL [90]), and so also the exact computation recommends itself for implementation.

Chapter 3

Piece-wise linear metrics

3.1 Preliminaries

In this chapter we will discuss medial axis computation for shapes with piecewise linear boundaries with respect to piece-wise linear metrics. The main focus will be on shapes in space, bounded by triangular meshes, for which an algorithm and its implementation is described.

Definition 3.1.1. *Throughout this chapter a shape* $\Omega \subset \mathbb{R}^d$ *for* $d = 2, 3$ *is considered to have a boundary* $\partial\Omega$ *which is piece-wise linear and triangulated. This means it consists of edges, vertices, and triangular facets (the latter ones only for* $d = 3$*), which we shall denote as* components *of this boundary. We shall refer to such a shape* Ω *as a* triangulated solid*.*

3.1.1 Piece-wise linear metric

In this section we will provide the basic setup of the underlying metric spaces.

Definition 3.1.2. *Consider a distance function* δ*. If the induced unit ball, as defined in Definition 1.2.2, is a polytope then we call* δ *a* piece-wise linear distance function*.*

The other way around, this allows us to derive a distance function from a d-polytope, which defines a special type of metric, called *quasi-metric* [85].

Definition 3.1.3. *A distance function* δ *defines a quasi-metric if it is positive definite and fulfills the triangle inequality, but is not necessarily symmetric.*

Definition 3.1.4. *Let* $\tilde{B}$ *be a bounded, open and convex polytope in* $\mathbb{R}^d$ *which contains the origin* O*. By assuming* $\tilde{B}$ *to be a unit ball, a distance function induced by* $\tilde{B}$ *for any two points* $x, y \in \mathbb{R}^d$ *can be defined as follows: Let* r *be the ray from* x *through* y *and* B *the unit ball* $\tilde{B}$ *translated by* $\overrightarrow{Ox}$*. There exists a unique intersection point* v *of* ∂B *and* r*. Then the distance function*

$$\delta_{\tilde{B}}(x, y) := \frac{\|y - x\|}{\|v - x\|} \tag{3.1}$$

defines a quasi-metric.[1]

---

[1] If $\tilde{B}$ is centrally symmetric with respect to the origin O, then $\delta_{\tilde{B}}$ is a *metric*.

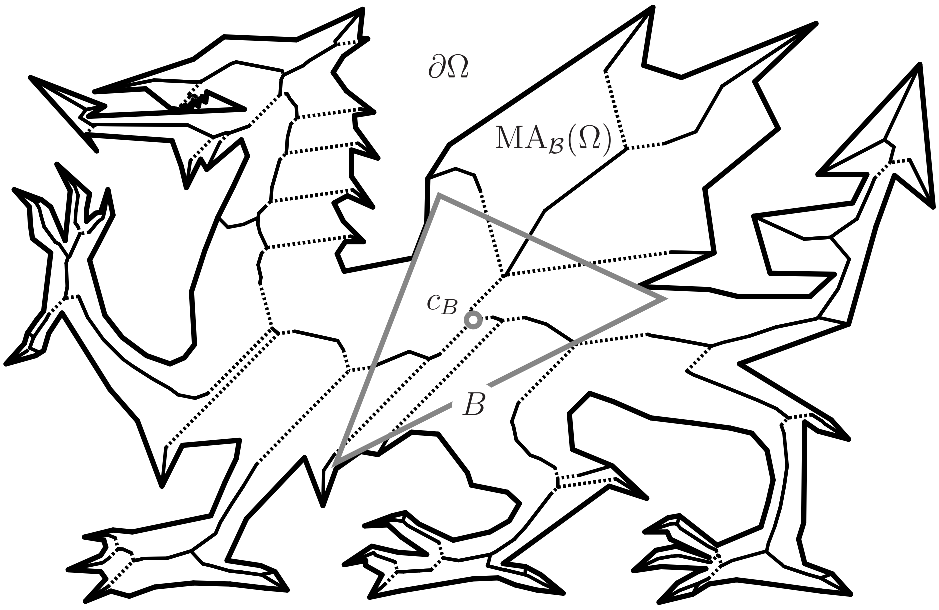


Figure 3.1: The picture shows a shape with its piece-wise linear boundary $\partial\Omega$ together with the medial axis $\mathrm{MA}_{\tilde{B}}(\Omega)$ (the dashed lines represent jump edges) induced by a piece-wise linear quasi-metric. The maximal* ball B is a translated and scaled copy of a polygonal unit ball $\tilde{B}$, and c_B the corresponding copy of the origin O.

For the remainder of this chapter, if not mentioned otherwise, O is considered to lie on the center of mass defined by the set of vertices $v_1, \cdots, v_k$ on $\partial\tilde{B}$:

$$O = \frac{1}{k} \cdot \sum_{i=1}^{k} v_i \ .$$

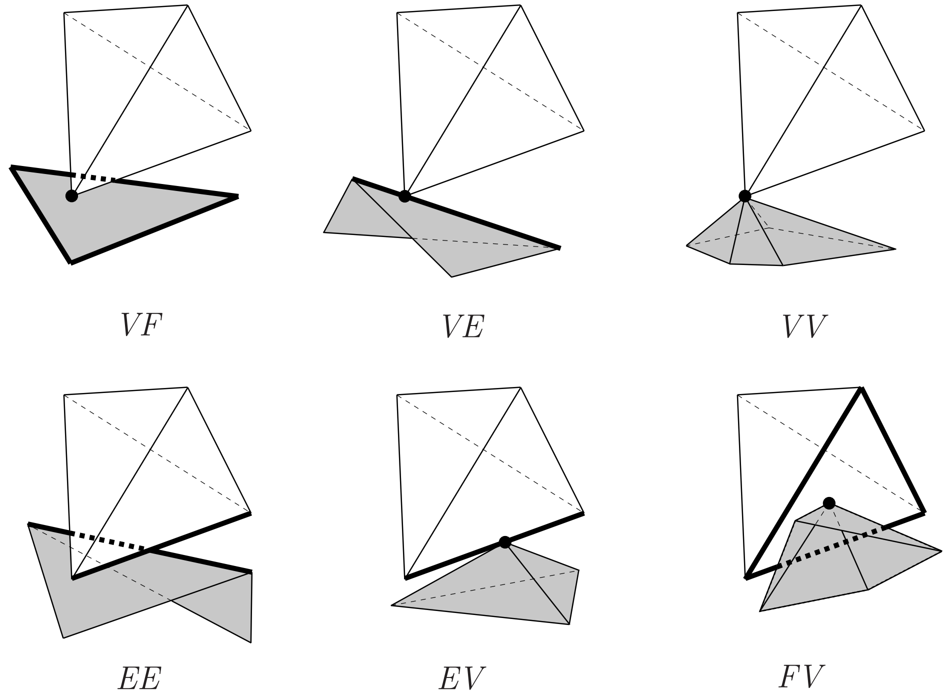
Furthermore we shall assume that no edge or facet of $\partial\tilde{B}$ is parallel to any edge or facet of $\partial\Omega$, or in other words, that Ω is in *general position* with respect to $\tilde{B}$. In space we also assume that $\partial\tilde{B}$ contains only triangular facets.

As introduced in Definition 1.2.2, the convex point set $B = B(x, \rho)$ is obtained from $\tilde{B}$ by applying restricted Euclidean similarity transformations consisting of a translation $\overrightarrow{Ox}$ combined with a scaling by the factor ρ. These convex sets represent *balls* with respect to the quasi-metric induced by $\tilde{B}$, since they consist of all points whose distance (with respect to $\delta_{\tilde{B}}$) from x does not exceed the scaling factor ρ.

In addition and as an alternative to the notion of maximal inscribed balls from Definition 1.2.3, the definition of another set of balls is essential when dealing with piece-wise linear quasi-metrics.

Definition 3.1.5. *A ball B is called a* maximal* ball *associated with Ω, if it is contained in Ω and its boundary ∂B shares at least two points with $\partial\Omega$. The set of all maximal* balls associated with Ω is called* $\mathrm{MAT}_{\tilde{B}}(\Omega)$.

In the Euclidean space, the two notions of maximal inscribed and maximal* are equivalent for piece-wise linear boundaries $\partial\Omega$. For piece-wise linear metrics and quasi-metrics however,



VF *VE* *VV*

EE *EV* *FV*

Figure 3.2: The six possible contact types for a polyhedral ball and a triangulated solid (boundary parts in grey), if they are in general position (which avoids the occurrence of contacts between a face of the unit ball and a face of the solid). The types *VF*, *EE* and *FV* are called *regular.*

the maximal inscribed balls are only a subset of the set of maximal* balls, as will be discussed in Section 3.1.3 and Section 3.1.4.

Definition 3.1.6. *The medial axis* $\mathrm{MA}_{\tilde{B}}(\Omega)$ *with respect to a piece-wise linear quasi-metric* $\delta_{\tilde{B}}$ *is defined as the union of the centers of all maximal* balls in* $\mathrm{MAT}_{\tilde{B}}(\Omega)$*.*

3.1.2 Contact types and classes

As said in Definition 3.1.1, the boundary of a triangulated solid can be grouped into vertices, edges and (in space) facets. The same applies to the boundary of a maximal* ball derived from a polygonal or polyhedral unit ball.

Since it is assumed that B and Ω are in general position, every boundary component (vertex, edge or facet) of a maximal* ball shares at most one point with a component of $\partial\Omega$. For a two-dimensional domain Ω in the plane, the following types of contact between $\partial\Omega$ and the boundary ∂B of a maximal* ball are possible:

VE: A vertex of ∂B is in contact with an edge of $\partial\Omega$,

VV: a vertex of ∂B is in contact with a vertex of $\partial\Omega$,

EV: or an edge of ∂B is in contact with a vertex of $\partial\Omega$.

The case where a maximal* ball possesses two contacts of the type EV that are realized at two different vertices of $\partial\Omega$, but only one edge of the ball, are excluded by requiring that no edge of ∂B is parallel to any line connecting any two vertices of $\partial\Omega$. (It suffices to assume that this condition is satisfied by all pairs of non-convex vertices of $\partial\Omega$.) This is subsumed by the fact that we assume B and Ω to be in *general position* and can be achieved easily by slight perturbation. This condition is necessary to avoid two-dimensional structures in the resulting medial axis.

In the three–dimensional case, the following types of contact between $\partial\Omega$ and the boundaries ∂B of maximal* balls are possible (for illustration see Figure 3.2):

VF: A vertex of ∂B is in contact with a facet of $\partial\Omega$,

VE: a vertex of ∂B is in contact with an edge of $\partial\Omega$,

VV: a vertex of ∂B is in contact with a vertex of $\partial\Omega$,

EE: an edge of ∂B is in contact with an edge of $\partial\Omega$,

EV: an edge of ∂B is in contact with a vertex of $\partial\Omega$,

FV: or a facet of ∂B is in contact with a vertex of $\partial\Omega$.

The case of maximal* balls with two contacts of the type FV which are realized in the interior of *only one facet* of ∂B are again excluded by assuming that Ω and B are in *general position.* The same applies to the case of two contacts of type *EE* which are formed by the same edge on ∂B, and whose shape component edges are lying on the same plane. Both cases would create three-dimensional components of the medial axis, which are clearly not wanted.

Observation 8. *The set of maximal* balls with two contacts of type EE which are realized in the interior of a single edge of the balls boundary cannot be avoided by perturbation, as the two shape edges do not have to lie on the same plane for such a constellation. The center points of these maximal* balls span a sheet which has no equivalent in the Euclidean case.*

Definition 3.1.7. *Consider a piece-wise linear quasi-metric. Then for a maximal* ball B associated with Ω, a footpoint is induced by a contact $C = (x, y)$, with x being the component of ∂B, and y of the shapes boundary $\partial\Omega$. The maximal* ball B is said to* feature *the contact C.*

A component x (vertex, edges, facet) from the boundary of a ball B is derived from a component $\tilde{x}$ of the respective unit balls boundary $\partial\tilde{B}$ by the corresponding similarity transformation. By this means, classes of contacts can be defined as follows.

Definition 3.1.8. *Given a shape Ω, a class of contacts $\mathcal{C} = (\tilde{x}, y)$ represents all contacts $C = (x, y)$ featured by maximal* balls associated with Ω, with x being derived from the primitive $\tilde{x} \subset \partial\tilde{B}$.*

By this means, a contact class can be associated with a set of maximal* balls.

Definition 3.1.9. *Given a shape Ω and a class of contacts $\mathcal{C} = (\tilde{x}, y)$, then the set of maximal* balls induced by $\mathcal{C}$ is denoted as follows:*

$$\mathcal{B}(\mathcal{C}) = \{B \mid B \textit{ is maximal}^{*} \textit{ and features a contact } (x, y)\} \ .$$

This allows a recursive definition for arbitrary many classes of contacts $\mathcal{C}_i$ with $i = 1 \dots n$:

$$\mathcal{B}(\mathcal{C}_1, \dots, \mathcal{C}_n) = \bigcap_{i=1}^{n} \mathcal{B}(\mathcal{C}_i) \ .$$

Observation 9. *For a contact class $\mathcal{C}$ of type VV, $\mathcal{B}(\mathcal{C})$ consists of a single maximal* ball.*

Definition 3.1.10. *Two contact classes $\mathcal{C}_1 = (\tilde{x_1}, y_1)$ and $\mathcal{C}_2 = (\tilde{x_2}, y_2)$ are said to be* neighboring *on $\partial\tilde{B}$, if*

$$\tilde{x_1} \cap \tilde{x_2} \neq \emptyset \ ,$$

and neighboring *on $\partial\Omega$ if the same applies to y_1 and y_2.*

So neighboring means, that two components share at least a vertex. A facet and an edge, for example, are neighboring, if the edge is part of the facets boundary or if one of the edges end-points is a corner point of the facet. An edge and a vertex are neighboring, if the vertex is one of the edges end-points, and so on.

Henceforth, in the plane, we shall call the contact types *VE* and *EV regular*, and the contact type *VV non-regular*. In space the contact types *VF*, *EE* and *FV* are considered *regular* and the contact types *VE*, *EV* and *VV non-regular*.

3.1.3 $\mathrm{MA}_{\tilde{B}}$ in the plane

First consider the two-dimensional case. The structure of the Euclidean medial axis $\mathrm{MA}(\Omega)$ for general shapes (and thus also for polygonal domains) is described in Section 1.2.

The axis $\mathrm{MA}_{\tilde{B}}(\Omega)$ with respect to a piece-wise linear metric $\delta_{\tilde{B}}$ has a similar structure:

Edge: An edge e of $\mathrm{MA}_{\tilde{B}}(\Omega)$ is formed by the center points of all maximal* balls which feature contacts from the same two classes of contacts $\mathcal{C}_i, \mathcal{C}_j$, being of the types *VE* or *EV*.

$$e = \{c_B \mid B \in \mathcal{B}(\mathcal{C}_i, \mathcal{C}_j)\} \ .$$

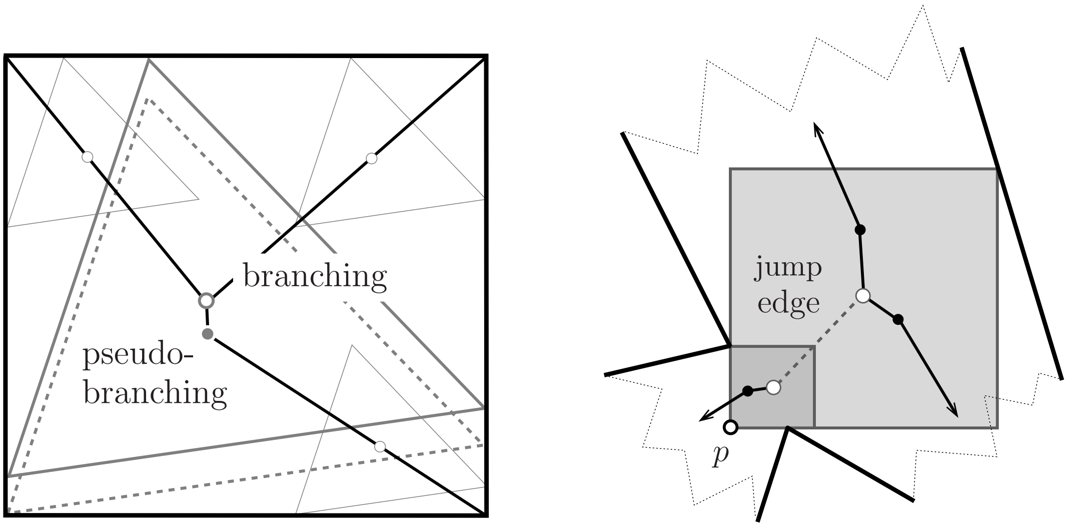
Branching point: A branching point of the medial axis is the center point of a maximal* ball featuring (in the non-degenerate case) three contacts.

Property 1. *For Ω a polygonal domain and $\tilde{B}$ a convex polygonal unit ball, the medial axis $\mathrm{MA}_{\tilde{B}}(\Omega)$ is piece-wise linear.*

Proof. The medial axis is composed of parts of bisectors between boundary components of Ω. As Ω is a triangulated solid, only bisectors between lines and points have to be considered, which are–when the underlying distance function is piece-wise linear– straight lines. As a consequence $\mathrm{MA}_{\tilde{B}}(\Omega)$ is piece-wise linear. □

Pseudo-branchings and jump edges

A new phenomenon that occurs when using a piece-wise linear metric instead of the Euclidean one is the implicit pruning of convex features of the boundary, which are flat with respect to the unit ball, in the sense that a vertex of the unit ball fits into the wedge defined by the feature. Such features lead, in 2D, to the appearance of special branching points (see Figure 3.3a), from now on called *pseudo branchings*. The maximal* ball centered there shares



(a) Pseudo-branching

(b) Jump edge

Figure 3.3: (*a*) A triangular unit ball induces a pseudo-branching in the medial axis of a square domain. (*b*) 2D example of a jump edge with center of scaling p.

only two points with the boundary of Ω, where one contact is of type VV and the other one regular. In contrast to a branching point, only two edges of the medial axis are incident to a pseudo-branching.

Another peculiarity for a piece-wise linear quasi-metric $\delta_{\tilde{B}}$ is that the centers of all maximal inscribed balls are not necessarily sufficient to obtain a connected medial axis $\mathrm{MA}_{\tilde{B}}(\Omega)$. Consider a maximal* ball B that features exactly two contacts of the contact classes $\mathcal{C}_i$ and $\mathcal{C}_j$ that are neighboring on the boundary of the unit ball (not on $\partial\Omega$) and let p be the point of incidence of the two unit ball components of these contact classes. In this case, any uniform scaling with center p and a factor f sufficiently close to 1 transforms B into another maximal* ball which is either a subset (if $f < 1$) or a super-set (if $f > 1$) of B. The latter means that B is not maximal inscribed. The edge $e = \mathcal{B}(\mathcal{C}_i, \mathcal{C}_j)$ of the medial axis which connects the possibly disconnected components induced by maximal inscribed balls is called a *jump edge* of $\mathrm{MA}_{\tilde{B}}(\Omega)$ since the maximal inscribed balls jump between the two extreme positions. See Figure 3.3b for an illustration.

A sum-up of the above:

Pseudo-branching: A pseudo-branching of $\mathrm{MA}_{\tilde{B}}(\Omega)$ is the center point of a maximal* ball featuring two contacts, where at least one of them is of type VV.

Jump edge: A jump edge is an edge of the medial axis, where the two defining contact classes are neighboring on $\partial\tilde{B}$.

3.1.4 $\mathrm{MA}_{\tilde{B}}$ in 3-space

In three dimensions, the medial axis of a triangulated solid with respect to a piece-wise linear quasi-metric consists of the following primitives:

Sheet: A sheet ς is the set of center points of maximal* balls which are defined by the same two classes of contacts $\mathcal{C}_i, \mathcal{C}_j$, being of regular type *VF*, *EE* or *FV*.

$$\varsigma = \{c_B \mid B \in \mathcal{B}(\mathcal{C}_i, \mathcal{C}_j)\} \ .$$

Seam: A seam s consists of the center points of maximal* balls featuring contacts from three classes of contacts $\mathcal{C}_i, \mathcal{C}_j, \mathcal{C}_k$, being of the same types as for the sheet.

$$s = \{c_B \mid B \in \mathcal{B}(\mathcal{C}_i, \mathcal{C}_j, \mathcal{C}_k)\} \ .$$

Junction point: If a maximal* ball features four (or more in the degenerate case) contacts of regular type, then its center defines a junction point of the medial axis.

By this means, in the general case, three sheets meet in a seam, and four seams are joined in a junction point (see again Figure 1.5b). Similar to the planar case the following can be stated:

Property 2. *For* Ω *a triangulated solid and* $\tilde{B}$ *a convex polyhedral unit ball, the medial axis* $\mathrm{MA}_{\tilde{B}}(\Omega)$ *is piece-wise linear.*

Proof. Similar as in the proof of Property 1 all bisectors with respect to a piece-wise linear distance function between boundary components of a triangulated solid are planes. The medial axis is composed of sheets which lie on these planes, and which are bounded by intersections of such planes, which are again linear. As such $\mathrm{MA}_{\tilde{B}}(\Omega)$ is piece-wise linear. $\square$

Pseudo-seams and jump sheets

The pseudo-structures in the three-dimensional case result from contact classes of the non-regular types *VE*, *EV* and *VV*. The center points of a set of maximal* balls, defined by two contact classes, where one of them is of the type *VE* or *EV*, form a component of $\mathrm{MA}_{\tilde{B}}(\Omega)$, that we will call henceforth *pseudo-seam*. While in a usual seam three sheets meet, only two sheets meet in a pseudo-seam (for illustration see Figure 3.14a and Figure 3.14b).

A point of the medial axis, which is the endpoint of at least one pseudo-seam is called a *pseudo-junction*. Let B be the maximal* ball with its center c_B at such a point, then it can be distinguished between three types of pseudo-junctions:

(I): Two pseudo-seams whose defining non-regular contact class are the same, and two seams meet at c_B. In this case B features three contacts, and one of them is of type *VE* or *EV*.

(II): Exactly four pseudo-seams are meeting at c_B, coming in pairs which share their regular defining contact class. In this case B features two contacts, where both are of type *VE* or *EV*.

(III): Several pseudo-seams with the same defining regular contact classe meet at c_B, what happens if and only if one of the two featured contacts of B is of type *VV*.

Similar to the discussion of jump edges in the planar situation one may observe that a maximal* ball B with only two contacts, whose classes are neighboring on $\partial\tilde{B}$, is not maximal inscribed, since it is possible to apply a uniform scaling with a center that is located in the intersection of the ball components of the two contacts. The center points of the set of maximal* balls obtained by such a scaling form a so-called *jump sheet* of the medial axis. A group of maximal* balls defined by three classes of contacts, which are all mutually neighboring on the boundary of the unit ball, and whose ball components all intersect in a common vertex, induces a *jump seam*, which naturally always lies on the boundary of a jump sheet.

A more formal definition is given in the following:

Pseudo-seam: A pseudo-seam is a part of $\mathrm{MA}_{\tilde{B}}(\Omega)$ that is defined by two classes of contacts, where one of the two classes is of the type VE or EV and the other one of a regular type.

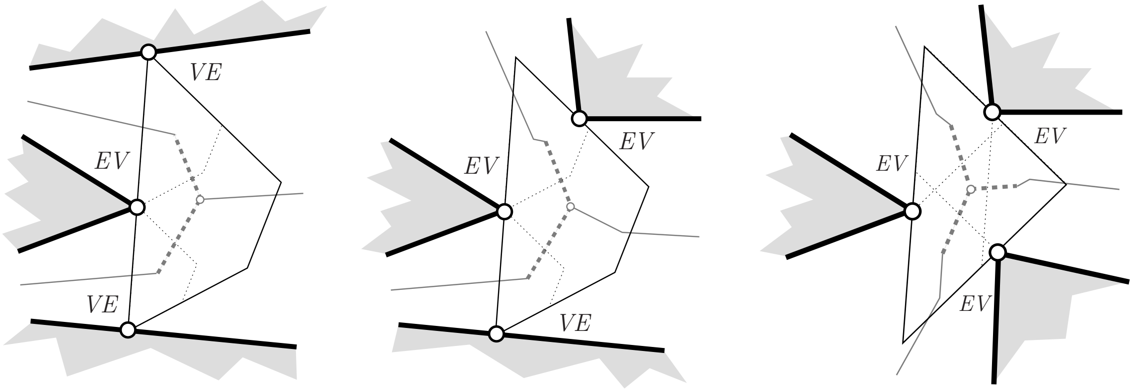
Pseudo-junction: A pseudo-junction is the center point of a maximal* ball with (I) three contacts, where one is of type VE or EV and the other two regular, (II) two contacts where both are of the type VE or EV, or (III) two contacts with one being of type VV and the other one regular.

Jump sheet: A jump sheet is a sheet of the medial axis, where the two defining contact classes are neighboring on $\partial\tilde{B}$.

Jump seam: A jump seam is a seam of the medial axis, defined by three mutually neighboring regular contact classes $(\tilde{x_1}, y_1)$, $(\tilde{x_2}, y_3)$ and $(\tilde{x_3}, y_3)$ such that

$$\tilde{x_1} \cap \tilde{x_2} \cap \tilde{x_3} \neq \emptyset \ .$$

Once again, the jump edges in 2D and jump sheets in 3D – and consequently the consideration of maximal* balls – are needed in order to guarantee that the medial axis of connected domains is again connected. Also note again, that the definition of maximal* would lead to two-dimensional components of the medial axis in the plane, and three-dimensional ones in 3-space, if the notion of general position would be relaxed.



(a) Two incident jump edges

(b) Three incident jump edges

Figure 3.4: Shown above are contact constellations, which cause branching points where several jump edges (grey dashed) join. The maximal* balls, indicated by dotted lines, are centered at the endpoint of a jump edge and feature one contact of type VV.

3.2 The medial axis of polygons

In this section we will discuss certain features of the piece-wise linear medial axis in the plane. Many of the properties that characterize this structure are easy to understand in the two-dimensional case, and can for the most part be intuitively extended to 3-space. As such the following can be seen as a warm-up for the three-dimensional set up.

3.2.1 Piece-wise and Euclidean medial axis

The medial axis $\mathrm{MA}_{\tilde{B}}(\Omega)$ with respect to a piece-wise linear metric $\delta_{\tilde{B}}$ suggests itself for comparison with the one for Euclidean distance (denoted by $\mathrm{MA}(\Omega)$). The two axes may differ from one another not only with regard to geometrical deviation, but also concerning structure and topology. Nevertheless a close connection between them can be established for certain preconditions on the unit ball, see Section 3.2.1.3 for more details.

3.2.1.1 Jumps

A jump edge e in the plane can only occur in correlation with a reflex vertex of $\partial\Omega$ which does induce a contact class of type EV.[2] A more formal definition is the following:

Observation 10. *Given a jump edge*

$$e = \{c_B \mid B \in \mathcal{B}(\mathcal{C}_i, \mathcal{C}_j)\} \ ,$$

at least one of the two contact classes $\mathcal{C}_i = (\tilde{x_i}, y_i)$ *and* $\mathcal{C}_j = (\tilde{x_j}, y_j)$ *must be of type EV.*

Proof. The reason for this is that the two contact classes $(\tilde{x_i}, y_i)$ and $(\tilde{x_j}, y_j)$ that define e have to be neighboring on $\partial\tilde{B}$. Assume that both contact classes are of type VE. Then the two

---

[2] One of the two maximal* balls centered at the endpoints of a jump edge has to feature a contact of type VV, where the shape component is a reflex vertex. Its center represents a pseudo-branching, while the center of the other ball is always a branching point of the axis.

vertices $\tilde{x_i}$ and $\tilde{x_j}$ have to be equal to be neighboring. Consider a maximal* ball B centered on e. It has to feature two different contacts where the ball component is the same vertex on ∂B. This would be at the joint vertex of y_i and y_j, defining a contact of type VV, with c_B being a pseudo-branching. Thus one of the two contact classes has to be of type EV. □

Consider a maximal* ball B, featuring the contacts C_1, C_2 and C_3, whose center represents a branching point of the medial axis. If the contact classes of C_1 and C_2 and the contact classes of C_2 and C_3 are neighboring on the unit balls boundary, then two jump edges meet at c_B (see Figure 3.4a). If even all three contact classes are mutually neighboring on $\partial\tilde{B}$, then three jump edges join at c_B. Note however that this is only possible for a triangular unit ball and three contacts of type EV as depicted in Figure 3.4b.

At least one of the two contact classes that induce a jump edge is of type EV. A jump edge can be seen as the replacement of all bisectors between the accordant reflex vertex on $\partial\Omega$ and any other boundary component of the shape in the Euclidean medial axis. By this means, jump edges in general introduce considerable structural and geometrical deviations between $\mathrm{MA}_{\tilde{B}}(\Omega)$ and $\mathrm{MA}(\Omega)$. It is obvious that, by trend, a combinatorially larger unit ball causes less jump edges, as featured contacts are less likely of contact classes that are neighboring on the boundary of $\tilde{B}$.

3.2.1.2 Implicit pruning

In the Euclidean case, the medial axis $\mathrm{MA}(\Omega)$ enters every convex feature (vertex) of a polygonal shape Ω. A maximal* polygonal ball B in the plane, which features a contact of type VV, induces a pseudo-branching of $\mathrm{MA}_{\tilde{B}}(\Omega)$. If this contact is realized at a convex vertex of $\partial\Omega$, then this corner is not visited by an edge of the medial axis induced by $\tilde{B}$.

Definition 3.2.1. *Consider a polygonal shape Ω, and a maximal* ball B, with respect to a quasi-metric $\delta_{\tilde{B}}$, which features a contact (x, y) of type VV. If y is a convex vertex of Ω, then* $\mathrm{MA}(\Omega)$ *does contain an edge which enters y, while* $\mathrm{MA}_{\tilde{B}}(\Omega)$ *does not. In this case we say that $\tilde{B}$* prunes the convex vertex *y (with respect to the Euclidean axis).*

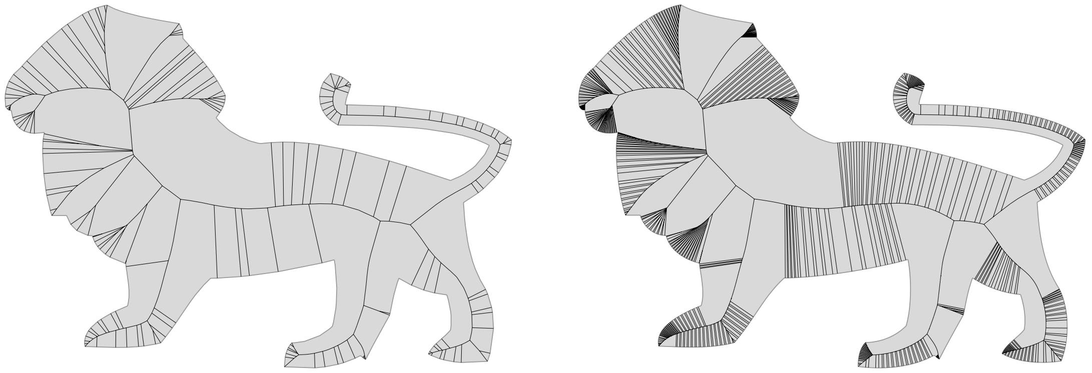
This *pruning* effect evokes structural differences between $\mathrm{MA}(\Omega)$ and $\mathrm{MA}_{\tilde{B}}(\Omega)$ concerning the number of branching points.

Corollary 5. *For a polygonal shape Ω, let m and $m_{\tilde{B}}$ be the number of branching points of* $\mathrm{MA}(\Omega)$ *and* $\mathrm{MA}_{\tilde{B}}(\Omega)$ *respectively. Let $l_{\tilde{B}}$ be the number of convex vertices on $\partial\Omega$ which are pruned by $\tilde{B}$, then*

$$m_{\tilde{B}} = m - l_{\tilde{B}} \ .$$

Proof. Every convex vertex on the boundary of Ω pruned by a polygonal unit ball $\tilde{B}$ induces a pseudo-branching in $\mathrm{MA}_{\tilde{B}}(\Omega)$ but a branching point in $\mathrm{MA}(\Omega)$. □

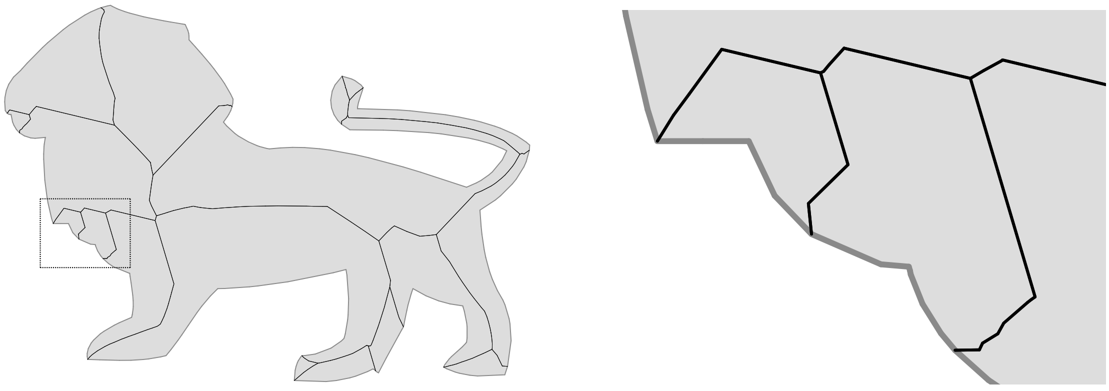
This technique of pruning now allows to thin out and capture the more essential parts of the medial axis by choosing an appropriate unit ball. This makes sense especially for approximations of shapes with via a larger number of segments. As shown in Figure 3.5a and Figure 3.5b the Euclidean medial axis can vary strongly for different approximations of the same shape, while the pruned piece-wise linear medial axis with respect to an appropriate polygonal unit ball is globally quite stable and deviates mainly locally (compare Figure 3.6 and Figure 3.7).



(a) 284 segments

(b) 1230 segments

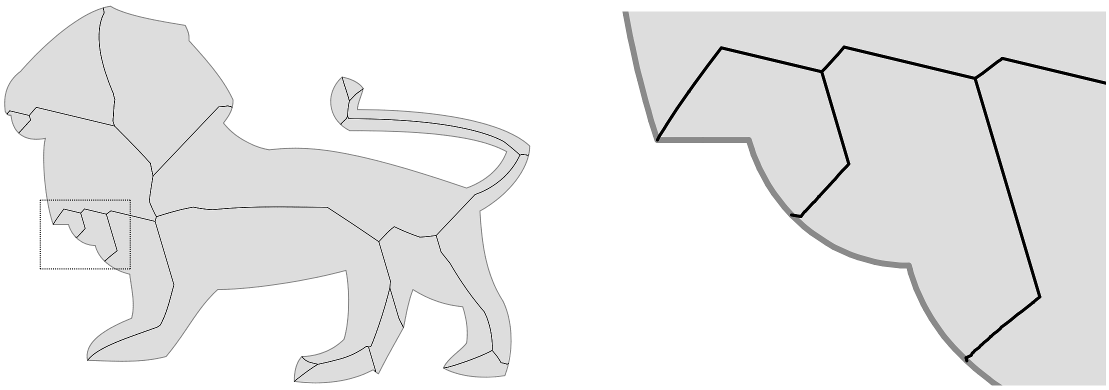
Figure 3.5: MA(Ω) for two input sizes of $\partial\Omega$.



(a) 284 segments

(b) Detail

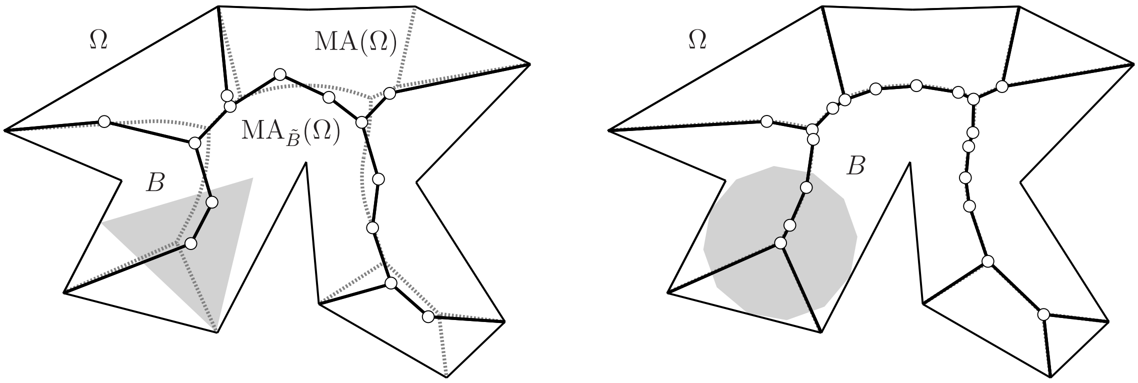
Figure 3.6: MA$_{\tilde{B}}(\Omega)$ for input size 284 and triangular $\tilde{B}$.



(a) 1230 segments

(b) Detail

Figure 3.7: MA$_{\tilde{B}}(\Omega)$ for input size 1230 and triangular $\tilde{B}$.



(a) $\tilde{B}$ is regular 3-gon

(b) $\tilde{B}$ is regular 12-gon

Figure 3.8: The medial axis of a simple polygon for different polyhedral unit balls.

3.2.1.3 Convergence

The quasi-metric defined by a convex polyhedron $\tilde{B}$ can be seen as an approximation of the Euclidean metric. Indeed, if the unit ball $\tilde{B}$ converges to the Euclidean unit ball, then the quasi-metric defined by it converges to the Euclidean metric. The convergence of the unit balls can be described with the help of the Hausdorff distance as defined in Definition 1.2.6.

In this section we consider simultaneously two metrics and the associated medial axes. On the one hand, there is the piece-wise linear (quasi-) metric $\delta_{\tilde{B}}$ defined by the convex polyhedron $\tilde{B}$ and the medial axis $\mathrm{MA}_{\tilde{B}}(\Omega)$ of the given domain Ω with respect to it. On the other hand, we have the usual Euclidean metric δ and the standard medial axis, which is denoted with $\mathrm{MA}(\Omega)$. The following result established a close connection between the two skeletal structures $\mathrm{MA}_{\tilde{B}}(\Omega)$ and $\mathrm{MA}(\Omega)$:

Theorem 2. *If the convex polyhedral unit ball $\tilde{B}$ that induces the (quasi-)metric $\delta_{\tilde{B}}$ converges to the Euclidean unit ball $\mathring{B}$, then the Hausdorff distance between the medial axes $\mathrm{MA}_{\tilde{B}}(\Omega)$ and $\mathrm{MA}(\Omega)$ of a triangulated solid Ω with respect to the piece-wise linear (quasi-)metric and the Euclidean metric, respectively, tends to zero:*

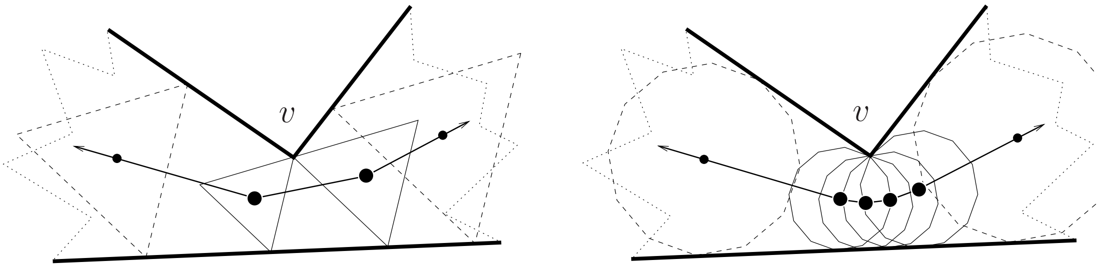
$$\mathrm{HD}(\tilde{B}, \mathring{B}) \to 0 \quad \Rightarrow \quad \mathrm{HD}(\mathrm{MA}_{\tilde{B}}(\Omega), \mathrm{MA}(\Omega)) \to 0 \ .$$

Thus, the convergence of the unit ball implies the convergence of the medial axis. See e.g. the example depicted in Figure 3.8a, where the piece-wise linear medial axis with respect to a triangular unit ball deviates notably from the (grey dashed) Euclidean medial axis, in particular concerning topology. The medial axis with respect to a regular 12-gon, as shown in Figure 3.8b, is considerably closer to the real medial axis $\mathrm{MA}(\Omega)$.

The detailed proof for the planar case can be found in [3].

3.2.2 Combinatorial size

As stated in Property 1, the medial axis of a polygonal area with respect to a piece-wise linear quasi-metric is also piece-wise linear. The number of line segments, which this axis is composed of, depends on various factors as



(a) $\tilde{B}$ is regular 3-gon (b) $\tilde{B}$ is regular 12-gon

Figure 3.9: The same reflex feature of a shape, handled with two polygonal unit balls of different combinatorial sizes.

1. the number of segments on the boundary of the shape Ω,
2. the geometrical alignment of $\partial\Omega$, in particular the number of its reflex features,
3. as well as the combinatorial size of the polyhedral unit ball $\tilde{B}$ and the placement of its defining vertices,

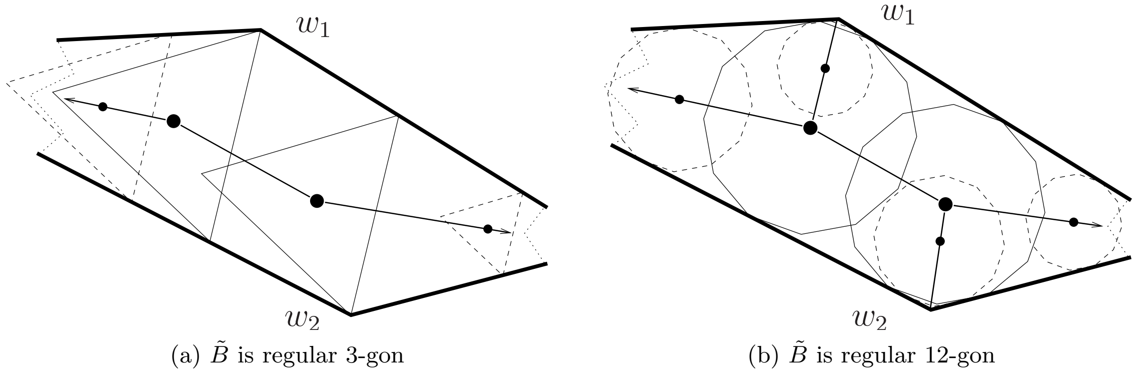
to name the most important ones. While it is trivially clear that the first point has a direct influence on the combinatorial size of the medial axis, the two other points are linked with one another and may require further explanation.

3.2.2.1 Reflex and convex features

Consider an edge e on $\partial\Omega$. The ball component x of every contact (x, e) featured by any maximal* ball, is derived from the same vertex $\tilde{x}$ on the unit balls boundary (as $\tilde{B}$ and Ω are considered in general position). This means, that for every edge on the boundary of the polygonal shape, there exists exactly one corresponding contact class of type *VE*. This is not necessarily true for vertices of $\partial\Omega$ which represent a reflex feature. For such a vertex v, there may exist several classes of contacts $(\tilde{x_i}, v)$ of type EV (with unit ball edges $\tilde{x_i}$), which define an edge of the medial axis. How many classes exist for a single reflex vertex depends on the unit ball's geometry and on its combinatorial size. As shown in Figure 3.9a, a regular 3-gon as unit ball creates one edge of the medial axis around the reflex vertex v. The use of a 12-gon (see Figure 3.9b) already results in 3 edges for the accordant portion of the axis.

Observation 11. *Consider a unit ball $\tilde{B}$ and a unit ball $\tilde{B}'$, where the vertices of $\tilde{B}$ are a subset of the vertices of $\tilde{B}'$ and a reflex vertex v on $\partial\Omega$. Then the number of different contact classes with v as their shape component induced by $\tilde{B}$ is smaller or equal to the number induced by $\tilde{B}'$.*

Also convex features have, depending on the unit ball, a certain influence on the combinatorial size of the medial axis. Consider a convex vertex w on $\partial\Omega$, being an endpoint of the two boundary edges e_1 and e_2. Each of these two edges has one corresponding contact



(a) $\tilde{B}$ is regular 3-gon (b) $\tilde{B}$ is regular 12-gon

Figure 3.10: The coarser unit ball to the left does not induce edges that lead into the flat convex features, what results in pseudo-branchings. The unit ball with size 12 does not prune these components.

class $\mathcal{C}_i = (\tilde{x_i}, e_i)$ with $\tilde{x_i}$ being a vertex of $\tilde{B}$, for $i = 1, 2$. If now $\tilde{x_1} = \tilde{x_2}$, then there exists a maximal* ball B that features a contact of the contact class $(\tilde{x_1}, w)$, whose center is thus a pseudo-branching. By this means, there is no edge of the medial axis which has w as an endpoint (see Figure 3.10a). For a different unit ball, the vertices $\tilde{x_1}$ and $\tilde{x_2}$ may be different, what results in a branching point and an additional edge of the medial axis, as shown in Figure 3.10b. So the choice of the unit ball does, via this pruning, also have a direct impact on the number of branching points, and thus structure of the resulting medial axis, as can also be seen in Figure 3.8a and Figure 3.8b.

Similar to the fact stated in Observation 11, a unit ball which contains the vertices of $\tilde{B}$ as a subset of its vertices causes the pruning of fewer or the same amount of edges as $\tilde{B}$.

3.2.2.2 An upper bound

For this section let n be the number of segments on $\partial\Omega$, r the number of its reflex vertices and k the number of segments on the boundary of the underlying unit ball $\tilde{B}$. When we denote the combinatorial size of the medial axis with $|\mathrm{MA}_{\tilde{B}}(\Omega)|$ then it is obvious that its size is at least linear with respect to n and thus $\Omega(n)$ is a lower bound for $|\mathrm{MA}_{\tilde{B}}(\Omega)|$.

For the upper bound, the variations in the medial axis size, caused by the unit ball choice as discussed above, have to be considered. In general it can be said that a combinatorially larger unit ball causes less pruning and more axis edges induced by reflex features. While the additional branches caused by a larger unit ball at convex features are in their number again at most linear to n, the number of segments generated by reflex vertices are in general dependent on k. Thus the following upper bound for the combinatorial size is obtained:

$$|\mathrm{MA}_{\tilde{B}}(\Omega)| \in \mathcal{O}(n + r \cdot k) \ .$$

Note that r can be replaced by the number of maximal reflex polygonal chains on $\partial\Omega$. A reflex polygonal chain is a connected chain of segments where all vertices, formed by two segments of the chain, represent a reflex corner of $\partial\Omega$. A reflex chain is maximal, if the two outer most endpoints of the chain form a convex corner on the boundary of Ω. Obviously, this bound is tight when considering the generalization of the example shown in Figure 3.11.

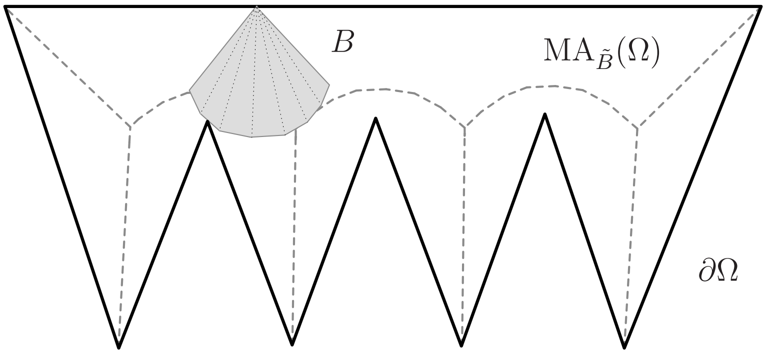
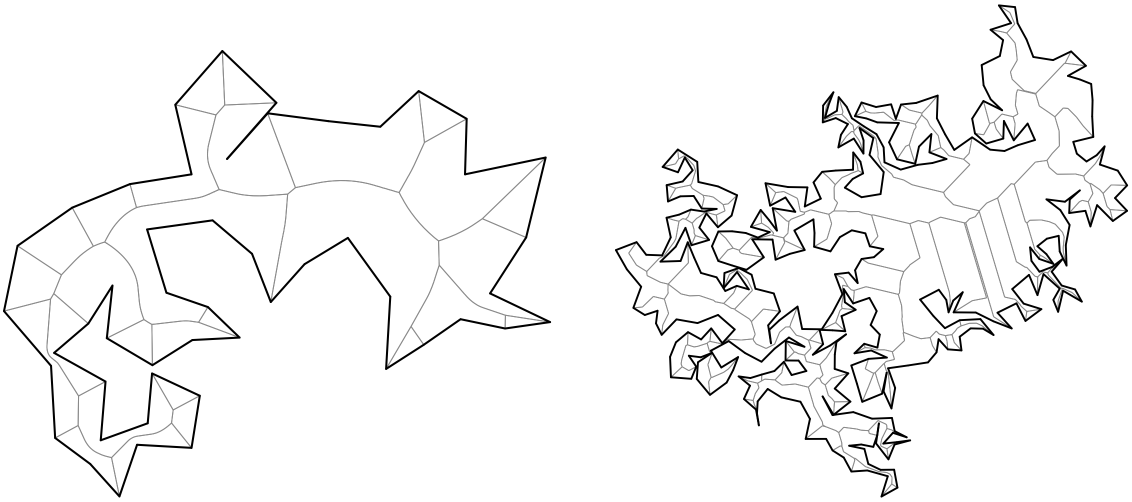


Figure 3.11: Such a *zigzag*-shape with size n and r reflex vertices together with a unit ball of size k as indicated by B induces a medial axis $\mathrm{MA}_{\tilde{B}}(\Omega)$ of size $\Theta(n + r \cdot k)$.



(a) $n = 50$, $k = 64$

(b) $n = 400$, $k = 3$

Figure 3.12: Two random polygons with their medial axis with respect to different unit balls.

Also for random polygons and regular k-gons as unit balls the connection between the size of the medial axis and k (disregarding the minimal effect of pruning) is clearly visible when consulting Table 3.1 and the according plot Figure 3.13. For this data, the medial axis $\mathrm{MA}_{\tilde{B}}(\Omega)$ has been computed for random polygons (for two examples see Figure 3.12) with different input sizes n, and for various unit balls $\tilde{B}$. All obtained values are averaged over 20 input polygons of the same combinatorial size. As can be seen from this experimental data, the size of the resulting axis is always between n and $n + rk$.

	3-gon		**12-gon**		**64-gon**		**200-gon**	
n	$\lvert\mathrm{MA}\rvert$	$^{\lvert\mathrm{MA}\rvert}/_{n+rk}$	$\lvert\mathrm{MA}\rvert$	$^{\lvert\mathrm{MA}\rvert}/_{n+rk}$	$\lvert\mathrm{MA}\rvert$	$^{\lvert\mathrm{MA}\rvert}/_{n+rk}$	$\lvert\mathrm{MA}\rvert$	$^{\lvert\mathrm{MA}\rvert}/_{n+rk}$
50	76	0.65	125	0.4	364	0.25	979	0.22
100	156	0.64	261	0.39	771	0.24	2095	0.22
200	315	0.64	522	0.38	1555	0.24	4226	0.21
400	634	0.64	1054	0.38	3145	0.24	8567	0.21
800	1265	0.64	2116	0.38	6305	0.24	17182	0.21
1600	2535	0.64	4239	0.38	12683	0.24	34555	0.21

Table 3.1: The sizes of the medial axis for various n and k – averaged over 20 input polygons – and the ratios with respect to the upper bound are given in the columns labeled with $\lvert\mathrm{MA}\rvert$ and $^{\lvert\mathrm{MA}\rvert}/_{n+rk}$ respectively.

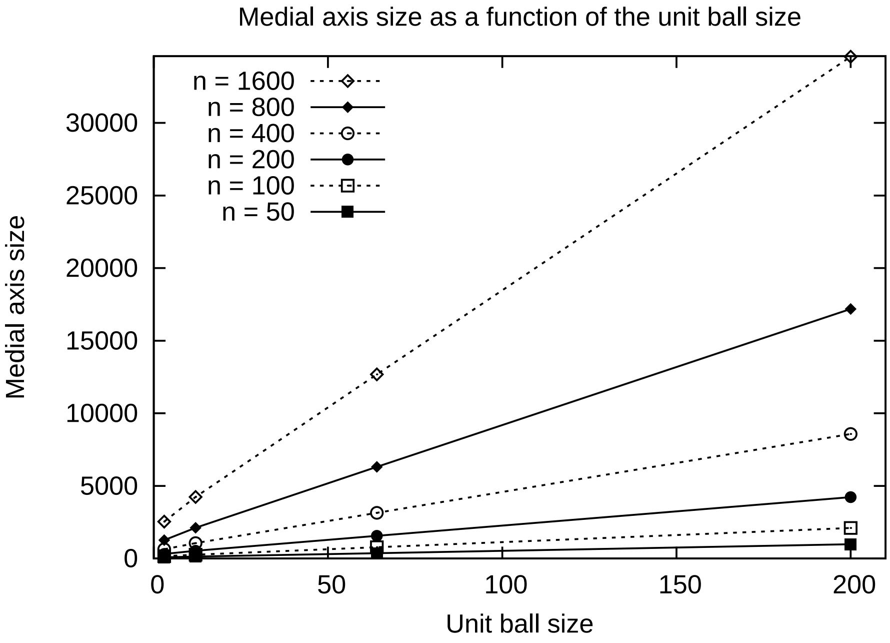
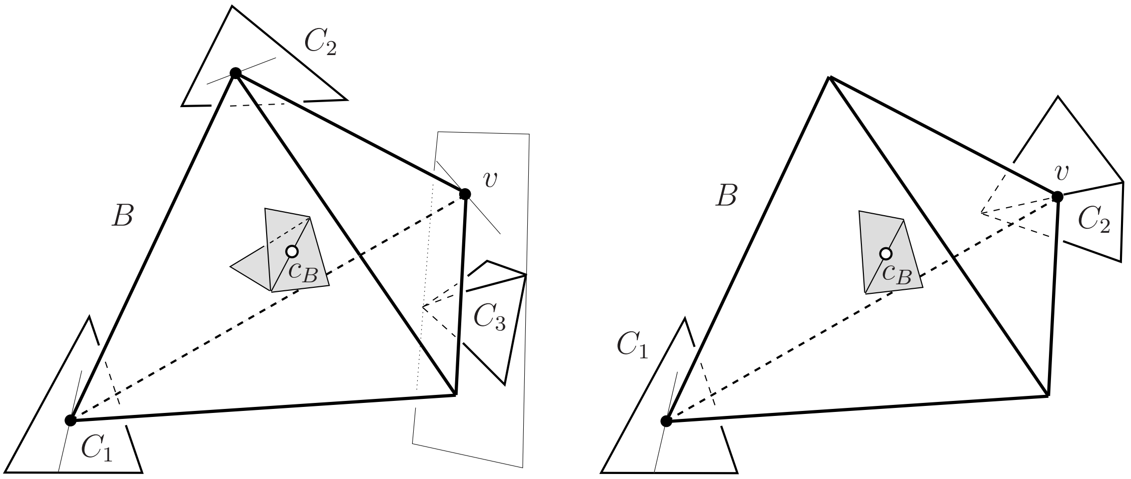


Figure 3.13: Plot of data from Table 3.1.



(a) Seam (b) Pseudo-seam

Figure 3.14: (a) The center c_B of the maximal* ball B lies on a seam of the axis. The point v, being the image of the representative vertex of $\mathcal{C}_3$, is its projection on the contact plane of $\mathcal{C}_3$. (b) The contact C_2 is of type VE as v lies on an edge of $\partial\Omega$. Therefore the center of B lies on a pseudo-seam, where only two sheets of the medial axis meet.

3.3 The medial axis of polyhedra

3.3.1 Projections and arrangements

In this section we will develop an algorithm for the medial axis computation in the three-dimensional case. Henceforth we consider solely shapes Ω which are triangulated solids, which means that they are bounded by a triangular mesh $\partial\Omega$. For such a solid we want to construct the medial axis with respect to a piece-wise linear quasi-metric $\delta_{\tilde{B}}$, induced by an arbitrary convex polyhedral unit ball. Note again, that $\mathrm{MA}_{\tilde{B}}(\Omega)$ is in this case also a piece-wise linear structure.

Recall that a ball B is a scaled and translated copy of the polyhedral unit ball $\tilde{B}$. For any boundary component $\tilde{x}$ of $\tilde{B}$, we denote with x its image under the restricted similarity transformation (translation and scaling) that maps $\tilde{B}$ to B. Moreover, for each boundary component $\tilde{x}$ of $\partial\tilde{B}$ we choose an arbitrary but fixed *representative* vertex $\tilde{v} = \tilde{v}(\tilde{x})$, which is one of the three vertices of a triangular facet, one of the two end points of an edge, or the vertex itself in the case of a vertex. Again, let v denote its respective copy on the boundary of B.

The regular contact types or combinations of boundary components – which determine the basic structure of the medial axis (sheets, seams and junctions) – are VF, EE and FV (see Section 3.1.4). Even for objects and unit balls in general position, maximal* balls featuring contacts of the non-regular types VE, EV and VV do occur, however they can be regarded as being *singular*, and do only induce the pseudo-structures of the medial axis.

Definition 3.3.1. *For a regular contact (x, y) of a maximal* ball, the component $y \in \partial\Omega$ and the ball component $x \in \partial B$ span a plane. This plane is the same for all contacts in the same contact class, and will be called the* contact plane *associated with the contact class.*

A non-regular contact class $(\tilde{x}, y)$ of type *VE* or *EV* in a similar way defines a contact line (parallel to y or $\tilde{x}$ respectively), a contact class of type *VV* a single point.

3.3.1.1 Projections

A maximal* ball B features at least two contacts, and unless its center defines a pseudo-junction of type (II), at least one of these contacts is regular (see Section 3.1.4). Every regular contact defines a contact plane, which contains the contact component of B.

Definition 3.3.2. *Let the contact (x, y), featured by the maximal* ball B, be of a regular type, and let $\tilde{v}(x)$ be the representative vertex of x. We call v, i.e., the equivalent of $\tilde{v}$ on the translated and scaled copy B of the unit ball $\tilde{B}$, the* projection *of the center c_B (or B) into the contact plane of $(\tilde{x}, y)$ (with respect to $\tilde{v}$).*

For a singular contact class of type *VE* or *EV*, the center point of a maximal* ball projects into the associated contact line. For a contact (x, y) of type *VV* the projection of the center is the vertex $y \in \partial\Omega$ itself. Note however, that – except for contacts of type *VF*, *VE* and *VV* – the footpoints of a maximal* ball do in general not coincide with the projections of their centers, as v does not have to be the point of contact.

Definition 3.3.3. *Let $\mathcal{C} = (\tilde{x}, y)$ be a contact class of regular type and $\tilde{v}$ its representative vertex. Then the set of all projections of the balls in $\mathcal{B}(\mathcal{C})$ into the contact plane of $\mathcal{C}$ describes a region in this contact plane. We will call this region*

$$\mathcal{D}(\tilde{x}, y) = \{v \mid \exists B \in \mathcal{B}(\mathcal{C}) \ : \ v \textit{ is projection of } c_B\}$$

the contact domain of $(\tilde{x}, y)$ (with respect to $\tilde{v}$).

Roughly speaking, the contact domain $\mathcal{D}(\tilde{x}, y)$ describes the trace of the representative vertex $\tilde{v}(\tilde{x})$ for all maximal* balls B which feature a contact represented by the contact class $(\tilde{x}, y)$. By this means a contact domain is a union of projections of medial axis sheets. As all sheets are polygonal subsets of planes, so are their projections on the contact plane of $\mathcal{C}$. Thus the contact domain is, as union of polygonal regions, again a domain with a piece-wise linear boundary.

For a contact class $(\tilde{x}, y)$ of type *VF* the contact domain is defined by the triangular facet y itself. For a *EE* or *FV* contact class, different choices of the representative vertex result in different contact domains, which naturally all lie on the same plane. For more details we refer to Section 3.3.2.2.2. The non-regular contact types do not define a two-dimensional domain at all.

3.3.1.2 Contact arrangements

A component γ of the medial axis is induced by the contact classes $\mathcal{C}_1, \ldots, \mathcal{C}_n$ with $2 \leq n \leq 4$,

$$\gamma = \{c_B \mid B \in \mathcal{B}(\mathcal{C}_1, \ldots, \mathcal{C}_n)\} \ .$$

Every regular typed contact class $\mathcal{C}_i = (\tilde{x}_i, y_i)$ defines a contact plane, and via its representative vertex a contact domain $\mathcal{D}_i = \mathcal{D}(\tilde{x}_i, y_i)$. The component γ now, representing a set of center points of maximal* balls, projects on the contact plane of $\mathcal{C}_i$, whereas the projection is contained in $\mathcal{D}_i$. Depending on γ, the projections may be of various forms.

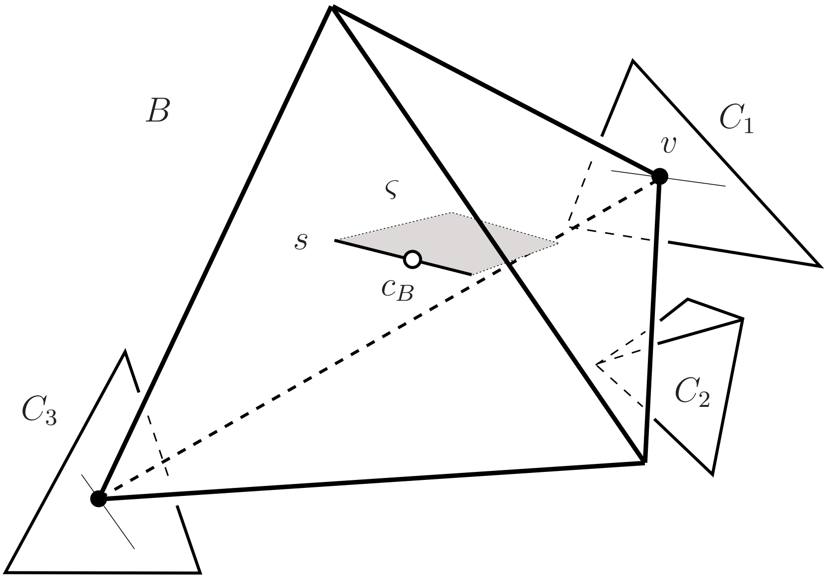


Figure 3.15: The center c_B of the maximal* ball B lies on the seam s. This seam lies on the boundary of the jump sheet ς, defined by the two contact classes represented by C_1 and C_2.

γ **is a sheet:** The projection of a sheet on the contact planes of both of its defining regular contact classes is a polygonal area.

γ **is a seam:** A seam is defined by three regular contact classes. Its projection on the three respective contact domains is a segment as illustrated in Figure 3.14a.

γ **is a junction:** A junction point projects on a single point inside the contact domains of all four defining contact classes.

γ **is a pseudo-seam:** A pseudo-seam projects to a segment on the contact plane of the one defining contact class, which is regular. The other, singular, contact class does not span a contact plane. See Figure 3.14b.

γ **is a pseudo-junction:** If a pseudo-junction has a regular defining contact class, than it projects down to a point inside the respective contact domain.[3]

γ **is a jump sheet:** Let $\mathcal{C}_1 = (\tilde{x_1}, y_1)$ and $\mathcal{C}_2 = (\tilde{x_2}, y_2)$ be the two defining contact classes of a jump sheet, and let

$$\tilde{c} = \tilde{x_1} \cap \tilde{x_2}$$

be the shared component on the boundary of the unit ball. Then the form of the projection depends on the choice of the representative vertex $\tilde{v_i}$ for the contact class $\mathcal{C}_i$, with $i = 1, 2$.

$$\gamma \text{ projects in } \mathcal{D}(\tilde{x_i}, y_i) \text{ to a } \begin{cases} \text{segment} & \text{if } \tilde{v_i} \subseteq \tilde{c} \\ \text{polygonal region} & \text{otherwise .} \end{cases}$$

---

[3] Pseudo-junctions of type (II) do not have a contact of regular type, however their projections lie on the boundary of two contact domains of regular typed contact classes.

See Section 3.3.2.4 for more details.

γ is a jump seam: The situation is similar to the one of jump sheets. Let $\mathcal{C}_i = (\tilde{x_i}, y_i)$ for $i = 1, 2, 3$ be the defining contact classes, and $\tilde{c}$ be the shared component on $\partial\tilde{B}$. Then for a contact class $\mathcal{C}_i$ and its representative vertex $\tilde{v_i}$

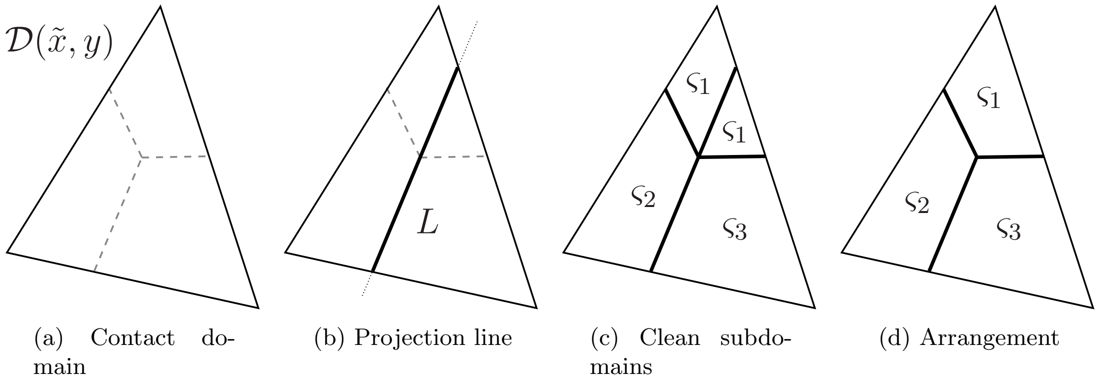
$$\gamma \text{ projects in } \mathcal{D}(\tilde{x_i}, y_i) \text{ to a } \begin{cases} \text{point} & \text{if } \tilde{v_i} \subseteq \tilde{c} \\ \text{segment} & \text{otherwise} \end{cases}$$

If a jump sheet projects to a segment in a contact domain, then the reason for this is that for all maximal* balls centered on this sheet, the point v derived from $\tilde{v}$ lies on a segment, which also contains the center of uniform scaling as explained in Section 3.1.4. By this means, all projections of ball centers into this domain (with respect to $\tilde{v}$) collapse down to this segment. Consider the example depicted in Figure 3.15. Here the seam s, defined by the respective contact classes of $C_i = (x_i, y_i)$ for $i = 1, 2, 3$, projects to a segment in all three affected contact domains. The contacts C_1 and C_2 share the vertex v on ∂B, and so their contact classes are neighboring on the unit ball, generating the jump sheet ς. If the vertex $\tilde{v}$ is also the representative vertex of the contact class of C_1, then ς projects into a segment on $\mathcal{D}(\tilde{x_1}, y_1)$. The projection of ς into $\mathcal{D}(\tilde{x_3}, y_3)$ is a polygonal domain.

Definition 3.3.4. *The projections of all seams, pseudo-seams – and possibly jump sheets and jump-seams for a certain choice of the representative vertex – that share a given contact class* $(\tilde{x}, y)$ *form an arrangement of line segments in the contact domain* $\mathcal{D}(\tilde{x}, y)$*. From now on this arrangement* $\mathcal{A}(\tilde{x}, y)$ *is called the* contact arrangement *(with respect to the representative vertex* $\tilde{v}(\tilde{x})$*).*

Every face of the contact arrangement represents a sheet of the medial axis. An edge of the arrangement represents either a seam or pseudo-seam, or a jump seam or jump sheet for an appropriate choice of the representative vertex. A vertex represents either a (pseudo-)junction, or possibly a jump seam. The edges meeting in such a vertex obey certain rules, reflecting the connectivity of the accordant medial axis components in space.

As a consequence, every components of the medial axis is in some way represented by a component of the contact arrangement of a contact class with regular type *VF*, *EE* or *FV* (see Section 3.1.4). In the following subsection we will describe how to obtain the contact domain for such a contact class, and provide an algorithm for the computation of the respective arrangement.



(a) Contact domain

(b) Projection line

(c) Clean subdomains

(d) Arrangement

Figure 3.16: Computation of a contact arrangement: (*a*) The contact domain is the intial subshape. (*b*) A crossing projection line L is found. (*c*) Each clean subdomain is part of the projection of a single medial axis sheet ς_i. (*d*) Subdomains defined by the same axis sheet are merged.

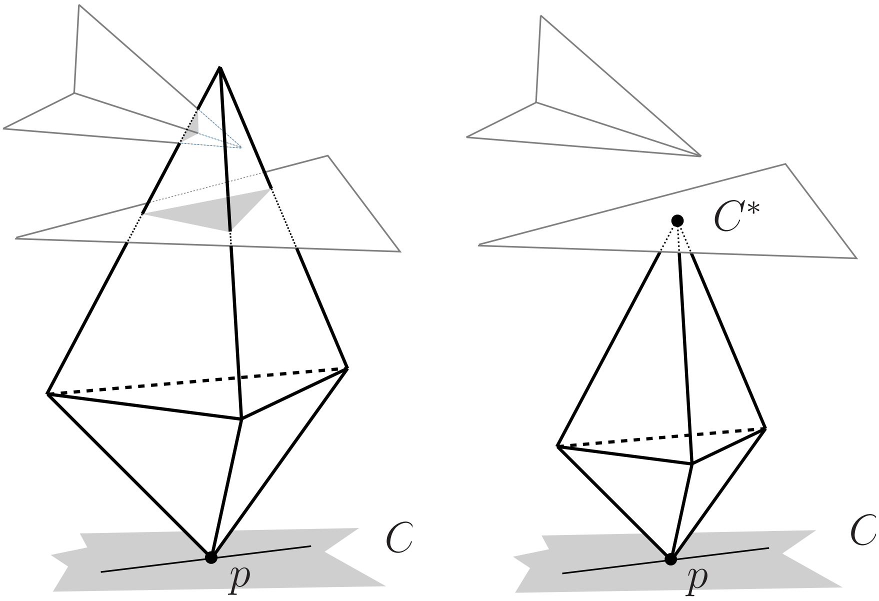
3.3.2 Arrangement Computation

For the triangulated solid Ω and the unit ball $\tilde{B}$ in general position, the center point of every maximal* ball with respect to Ω projects into the contact domain of at least one contact class of regular type *VF*, *EE* or *FV*. By this means, the medial axis $\mathrm{MA}_{\tilde{B}}(\Omega)$ is fully represented by the contact arrangements of all regular typed contact classes. In order to analyze the medial axis, the contact arrangements for all relevant contact classes $(\tilde{x}, y)$ are computed. Consequently, the problem of medial axis computation in space is reduced to a finite number of two-dimensional problems, which can moreover be addressed in parallel, since they are mutually independent.

3.3.2.1 Algorithm outline

For each contact class $\mathcal{C} = (\tilde{x}, y)$ of regular type, the contact domain $\mathcal{D}(\tilde{x}, y)$ with respect to the representative vertex is computed (details in Section 3.3.2.2.2). Afterwards the following algorithm, summarized visually in Figure 3.16, is performed:

1. Create a stack of polygonal subdomains in the contact plane of $\mathcal{C}$ and initialize it with the entire contact domain $\mathcal{D}(\tilde{x}, y)$.
2. If the stack is empty, then continue with step 4, otherwise take a subdomain D from the stack.
3. Check if there exists a seam, pseudo-seam or (for an appropriate representative vertex) jump seam or jump sheet, which defines a projection segment in $\mathcal{D}(\tilde{x}, y)$ that hits the subdomain D. If such a projection is found, then split D along the projection line L, being the supporting line of the projection segment, into two new subdomains and add them to the stack. Continue with step 2.
4. Remove all redundant line segments in the arrangement that do not represent projections of seams, pseudo-seams or jump sheets.



(a) Ball before shrinking (b) The maximal* ball at p

Figure 3.17: Iterative computation of a maximal* ball for a given projection p in the contact plane associated with C. The opposite contact is C^*.

3.3.2.2 Constructing maximal* balls

Let $\tilde{v}$ – chosen from $\tilde{x}$ – be the representative vertex of the contact class $\mathcal{C} = (\tilde{x}, y)$. The most frequent (and also most expensive) operation of the algorithm is to compute a maximal* ball B for a point p on the contact domain of $\mathcal{C}$, such that $v \in \partial B$ and p coincide. In particular, it is crucial to identify the remaining contacts of such a maximal* ball, as these contacts decide which part of the medial axis projects to p. If there is only one additional contact of regular type, then p lies on a face of the contact arrangement, otherwise, it belongs to an edge or vertex.

A maximal* ball is found by iterative shrinking – similar to the procedure described for piece-wise circular boundaries in Algorithm 1 – where p is the center of scaling. The starting ball has to satisfy $p = v$ and has to be sufficiently large to intersect the boundary mesh (see Figure 3.17a). With help of an *AABB (Axis Aligned Bounding Box) tree* (see next section), the intersections between components of the balls boundary and the mesh are efficiently detected. The component of the mesh closest to p determines the shrinking factor. This is done iteratively until the shrunk ball and the mesh are intersection-free (see Figure 3.17b). The last component of the mesh which is used to define the shrinking induces the second contact of the maximal* ball, from now on called *opposite* contact.

Lemma 8. *If all vertices of $\partial\Omega$ and $\partial\tilde{B}$ have rational coordinates, then all vertices of a maximal* ball B have rational coordinates, if at least one of them does.*

Proof. Let u be the vertex with rational coordinates, and let $w \neq u$ be a vertex of B which lies on the ball component of an arbitrary contact $(\tilde{x}, y)$ featured by B. The shape component y is rational, and the line parallel to the vector $\overrightarrow{\tilde{u}\tilde{w}}$ and containing u is rational. As such also

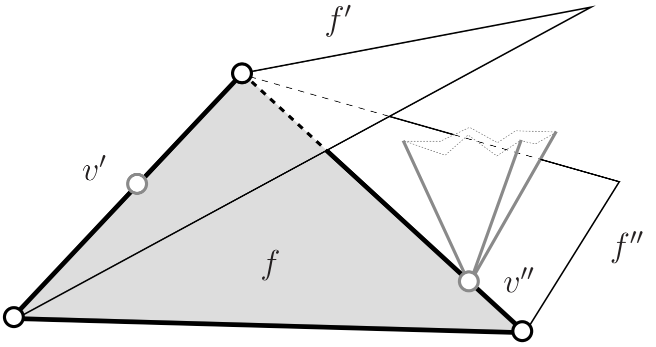


Figure 3.18: The maximal* ball projecting on v' is of size zero, as f forms a convex edge with f'. The ball projecting on v'' is centered on a pseudo-seam.

their intersection, being w, has rational coordinates. The rational factor $\frac{\|\tilde{u}-\tilde{w}\|}{\|u-w\|}$ is the same for all pairs of points on $\partial\tilde{B}$ and ∂B, therefore all vertices of B have rational coordinates. $\square$

This means that all computations needed for the construction of maximal* balls can be done within the set of rational numbers, if p on the contact plane is chosen rational. Note that also the center points of such maximal* balls are rational, and as such span components of the medial axis which are rational.

3.3.2.2.1 AABB Trees

We use the Axis Aligned Bounding Box tree [9], provided by the CGAL library [90] for fast detection of intersections between $\partial\Omega$ and ∂B for a maximal* ball B. The AABB tree is mainly a hierarchy of bounding boxes. A more detailed explanation is given in [9]:

> The AABB tree component offers a static data structure and algorithms to perform efficient intersection and distance queries against sets of finite 3D geometric objects. The set of geometric objects stored in the data structure can be queried for intersection detection, intersection computation and distance.
> ...
> Examples of intersection queries include line objects (rays, lines, segments) against sets of triangles, or plane objects (planes, triangles) against sets of segments.
> ...
> The AABB tree data structure takes as input an iterator range of geometric data, which is then converted into primitives. From these primitives a hierarchy of axis-aligned bounding boxes (AABBs) is constructed and used to speed up intersection and distance queries.

The triangles of the static shape boundary mesh are stored in the AABB tree, and are checked for intersections with the boundary triangles of a maximal* ball. The performance benefit of the tree is hard to measure in terms of bounds, as it strongly depends on the form and alignment of the boundary of the shape Ω. However, considerable speed-up can be measured for large mesh data with roughly equally sized triangles as demonstrated in Section 3.3.4.2.

3.3.2.2.2 Computing the contact domain

As defined in Definition 3.3.3, the contact domain of a contact class $\mathcal{C} = (\tilde{x}, y)$ (with respect to the representative vertex $\tilde{v}$) is the locus of all points p on the contact plane of $\mathcal{C}$ which have the property that there exists a maximal* ball $B \in \mathcal{B}(\mathcal{C})$ such that B projects on p (meaning that the accordant copy v of the representative vertex coincides with p).

The polygonal boundary of the contact domain is needed for initialization of the main algorithm as discussed in Section 3.3.2.1. This is trivial for a contact class $(\tilde{v}_b, f_s)$ of type *VF*, as the representative vertex $\tilde{v}$ is identical to the ball component $\tilde{v}_b$, and thus the domain is equivalent to the facet being the shape component. The set of balls projecting on an edge of the boundary of such a contact domain $\mathcal{D}(\tilde{v}_b, f_s)$ either induce a pseudo-seam of $\mathrm{MA}_{\widetilde{B}}(\Omega)$, or they are all of size zero if the corresponding edge on $\partial\Omega$ is convex with respect to $\widetilde{B}$. See Figure 3.18 for an illustration. Similar to this, a ball projecting on a vertex on the boundary either defines a pseudo-junction, or is of size zero if the vertex is convex.

For a contact class of type *EE* or *FV* the appearance of a contact domain is dependent on the choice of $\tilde{v}$, and as such has a more complex form. However, what can be stated for all contact domains of regular typed contact classes is the following:

Observation 12. *Consider a contact class $\mathcal{C} = (\tilde{x}, y)$ of regular type with representative vertex $\tilde{v}$ and contact plane $\mathcal{P}_\mathcal{C}$. Furthermore consider another contact class $\mathcal{C}' = (\tilde{x}', y')$ for which it is true that*

- *$\mathcal{C}'$ is of non-regular type and*
- *$y' \subseteq y$.*

Then the projection (with respect to $\tilde{v}$) of a maximal ball, featuring a contact of $\mathcal{C}'$, into $\mathcal{P}_\mathcal{C}$ lies on the boundary of $\mathcal{D}(\tilde{x}, y)$.*

3.3.2.2.3 Contact domain for *EE* contacts

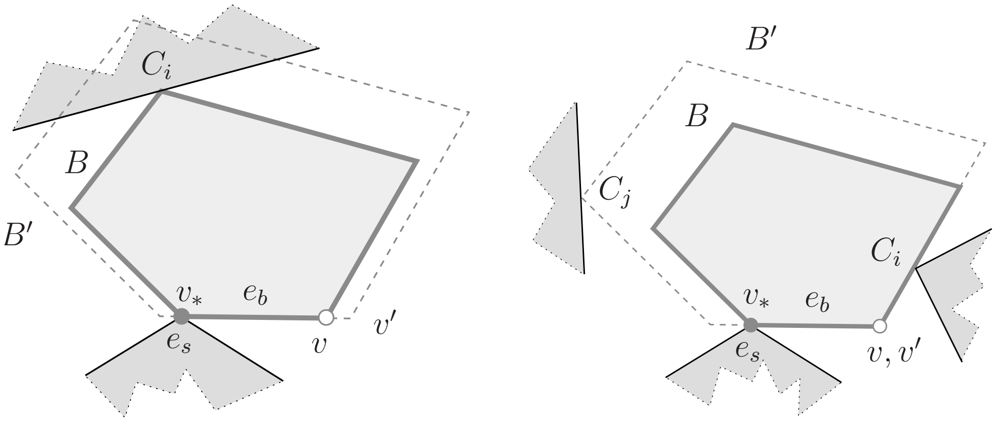
Consider a contact class $\mathcal{C} = (\tilde{e_b}, e_s)$ of type *EE* and its contact plane $\mathcal{P}_\mathcal{C}$. Let $\tilde{v} = \tilde{v}(\tilde{e_b})$ be the representative vertex of $\mathcal{C}$, and $\tilde{v_*}$ the other endpoint of $\tilde{e_b}$.

Proposition 1.

1. *The contact domain $\mathcal{D}(\tilde{e_b}, e_s)$ is contained in one of the two halfplanes in $\mathcal{P}_\mathcal{C}$, separated by the supporting line of e_s.*
2. *The edge e_s lies on the boundary of $\mathcal{D}(\tilde{e_b}, e_s)$.*

Proof. Ad 1: For every contact (e_b, e_s) of the class $\mathcal{C}$, the endpoint $v \in \mathcal{P}_\mathcal{C}$ lies on one side of the supporting line of e_s and v_* on the other. So the choice of the representative vertex $\tilde{v}$ dictates, which halfplane contains the contact domain (again with respect to $\tilde{v}$).
Ad 2: The edge e_s represents in $\mathcal{P}_\mathcal{C}$ the projection of all maximal* balls in $\mathcal{B}(\mathcal{C}')$ for $\mathcal{C}' = (\tilde{v}, e_s))$. The class $\mathcal{C}'$ is of non-regular type and $e_s \subseteq e_s$, see Observation 12. □

Proposition 2. *The contact domain $\mathcal{D}(\tilde{e_b}, e_s)$ is contained in the slat of $\mathcal{P}_\mathcal{C}$ that is bound by the two lines being parallel to the supporting line of $\tilde{e_b}$, and containing one endpoint of e_s respectively.*



(a) $v \neq v'$

(b) $v = v'$

Figure 3.19: In the pictures above two sketches of a cross section through the shape Ω and a maximal* ball B are shown, where the edge $e_s \in \partial\Omega$ lies perpendicular to the viewer-plane. While in the left picture the contact (e_b, e_s) is not neighboring to C_i on the ball, it is in the right one.

Proof. For every contact (e_b, e_s) of the class $\mathcal{C}$, e_b must intersect e_s. For a fixed e_s this is only true if e_b, and thus v lies in the slat as defined above. $\square$

Proposition 3. *Consider the contact class* $\mathcal{C}_* = (\tilde{v_*}, e_s)$ *of type VE. Then the projection (with respect to* $\tilde{v}$*) of all balls in* $\mathcal{B}(\mathcal{C}_*)$ *in* $\mathcal{P}_\mathcal{C}$ *forms a polygonal line, which is part of the boundary of* $\mathcal{D}(\tilde{e_b}, e_s)$*.*

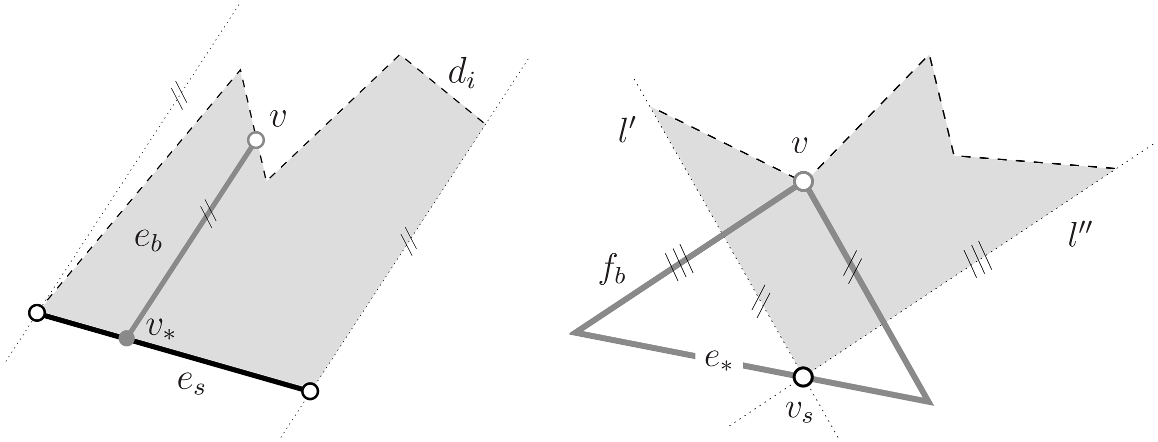
Proof. Consider a maximal* ball $B \in \mathcal{B}(\mathcal{C}_*)$ and the corresponding copy v of the representative vertex. Assume that v does not lie on the boundary of $\mathcal{D}(\tilde{e_b}, e_s)$. This means, that there exists a maximal* ball $B' \supsetneqq B$ which shares the footpoint of B on e_s, but with $v' \neq v$. As can be seen in Figure 3.19a, this is a contradiction to the fact that B is maximal*. Thus v has to lie on the boundary of $\mathcal{D}(\tilde{e_b}, e_s)$. $\square$

Every segment d_i of the polygonal line described in Proposition 3 corresponds to the projection of a set $\mathcal{B}(\mathcal{C}_*, \mathcal{C}_i)$ of maximal* balls. If $\mathcal{C}_i$ is neighboring to $\mathcal{C}$ on $\partial\tilde{B}$ then d_i is even the projection of a jump sheet and consequently of a seam represented by $\mathcal{B}(\mathcal{C}, \mathcal{C}_i, \mathcal{C}_j)$ on $\mathcal{P}_\mathcal{C}$ as illustrated in Figure 3.19b.

In Figure 3.20a an example for a possible contact domain of a contact class of type *EE* is given. To find the points shared between the segments of the polygonal line, a procedure based on the `projline` algorithm as introduced in Section 3.3.2.3.3 is used.

3.3.2.2.4 Contact domain for FV contacts

Let $\mathcal{C} = (\tilde{f_b}, v_s)$ be a contact class of type FV and $\mathcal{P}_\mathcal{C}$ its associated contact plane. Let $\tilde{v}$ be the representative vertex on the triangle represented by $\partial\tilde{f_b}$, and $\tilde{e_*}$ the segment on this triangle that is not incident to $\tilde{v}$. Then, similar to the contact domain for contact classes of type *EE*, certain criteria can be given for the placement and form of $\mathcal{D}(\tilde{f_b}, v_s)$:



(a) EE contact domain (b) FV contact domain

Figure 3.20: Two contact domain examples.

1. The contact domain of $\mathcal{C}$ has v_s on its boundary.
2. Consider the two supporting lines of the segments on $\partial\tilde{f}_b$ that are incident to $\tilde{v}$ and call them $\tilde{l}'$ and $\tilde{l}''$. Let l' and l'' be the lines through v_s that are parallel to $\tilde{l}'$ and $\tilde{l}''$ respectively. Then $\mathcal{D}(\tilde{f}_b, v_s)$ does overlap neither l' nor l''.
3. Let $\mathcal{C}_*$ be the contact class $(\tilde{e_*}, v_s)$. Then the polygonal line in $\mathcal{P}_\mathcal{C}$, resulting from the projection of all maximal* balls in $\mathcal{B}(\mathcal{C}_*)$ is part of the boundary of $\mathcal{D}(\tilde{f}_b, v_s)$.

Ad 1: The reflex vertex v_s of $\partial\Omega$ induces a maximal* ball featuring the contact (v, v_s) of type VV. The projection of this ball on $\mathcal{P}_\mathcal{C}$ (with respect to $\tilde{v}$) is v_s. As $v_s \subseteq v_s$, v_s lies, according to Observation 12, on the boundary of the domain.
Ad 2: The two lines l' and l'' split the contact plane into four sectors. $\mathcal{D}(\tilde{f}_b, v_s)$ has to be contained as a whole in one of these sectors.
Ad 3: The obtained polygonal line also marks the sector which contains the contact domain. See Figure 3.20b for a detailed example.

3.3.2.3 Finding projection lines

Given a contact class $\mathcal{C} = (\tilde{x}, y)$, consider its assigned final contact arrangement. Two maximal* balls projecting into two different faces of this arrangement feature of course both a contact of $\mathcal{C}$, but their opposite contact comes from two different contact classes. By this means, a projection line in a contact domain – determined by a seam (possibly combined with a jump sheet) or a pseudo-seam – always corresponds to a change of the opposite contact of the associated maximal* balls. Thus, the projection lines subdivide a contact domain into subdomains D_i whose points are the projections of maximal* balls with the same opposite contact.

Consider two points p and q on the contact domain of $\mathcal{C}$, which both do not lie on a projection line. If the two associated maximal* balls B_p and B_q feature opposite contacts C_p and C_q of two different contact classes $\mathcal{C}_p$ and $\mathcal{C}_q$, then there exists a projection line crossing

$\overline{pq}$. On the other hand, if $\mathcal{C}_p = \mathcal{C}_q$, then this does *not* imply that there is no such crossing line, since the faces of the final contact arrangement are not necessarily convex.

3.3.2.3.1 Different opposite contact classes

If the associated opposite contacts C_p and C_q of p and q are of different contact classes, a point on the segment $\overline{pq}$ has to be found, which lies on a projection line. We now assume that $\mathcal{C}_p \neq \mathcal{C}_q$, and that exactly one projection line L exists, that crosses $\overline{pq}$. Then three categorical possibilities for the change of the opposite contact and the resulting L exist (for illustrating sketches see Figure 3.21):

(a): The contact classes $\mathcal{C}_p$ and $\mathcal{C}_q$ are neighboring on $\partial\tilde{B}$ and on $\partial\Omega$. Then the projection of a pseudo-seam crosses $\overline{pq}$.

(b): There exists a maximal* ball, projecting on $\overline{pq}$, that features a contact of $\mathcal{C}_p$ and $\mathcal{C}_q$. The ball is centered on a seam, and its projection crosses $\overline{pq}$.[4]

(c): There exists a maximal* ball, projecting on $\overline{pq}$, that features a contact of one of the two known contact classes, let's say $\mathcal{C}_p$. Furthermore it features a contact of a contact class $\mathcal{C}'_q$ which is neighboring to $\mathcal{C}_q$, on the boundary of both, the unit ball and the shape. Then this ball is centered on a seam, that lies on the boundary of a jump sheet. The seam projects on a line in the contact plane of $\mathcal{C}$, which crosses $\overline{pq}$. For a certain choice of the representative vertex, also the jump sheet projects on this line, causing this special possibility for a change of the opposite contact. For another representative vertex the projection of the jump sheet is two-dimensional and the bounding projections of seams and pseudo-seams are found as indicated in case (a) and (b). See Section 3.3.2.4 for details on this.

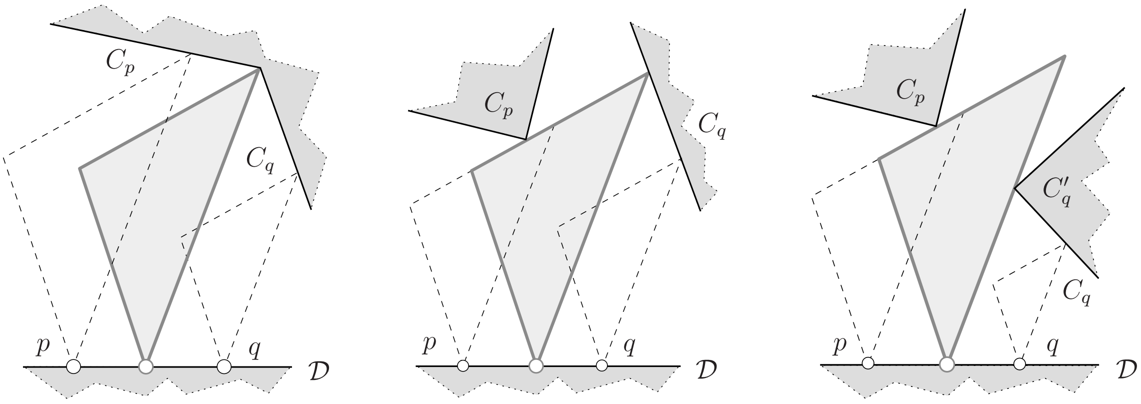
Henceforth we shall say that $\mathcal{C}_p$ and $\mathcal{C}_q$ *are witnesses* to a projection line that crosses $\overline{pq}$, if they fit one of the three scenarios described above.

If the two different contact classes are not witnesses to a projection line, then more than one projection line crosses $\overline{pq}$. This means that a point t on $\overline{pq}$ exists, whose associated maximal* ball features a opposite contact that is neither in $\mathcal{C}_p$ nor in $\mathcal{C}_q$. Such a point is computed by the simple binary search algorithm `searchcc`, which takes the two points p, q as well as $\mathcal{C}$ as arguments (see Algorithm 3).

By applying `searchcc` recursively, a pair of points on $\overline{pq}$ can be found, whose opposite contact classes are witnesses to a projection line that crosses $\overline{pq}$. In non-degenerate configurations, this process is guaranteed to terminate. For an illustrative example have a look at the two points p and q in the final contact arrangement $\mathcal{A}$ as depicted in Figure 3.22a. Their associated opposite contact classes are different, but they are not witnesses to a projection line, as more than one line crosses $\overline{pq}$. The segment $\overline{pt}$, with t being the midpoint of p and q, is still crossed by two projection lines. Finally, `searchcc`(p, t, $\mathcal{C}$) returns the point t', whose opposite contact class witnesses together with $\mathcal{C}_p$ a single projection line.

---

[4] Possible jump seam projections are indicated in the same way.



(a) Pseudo-seam (b) Seam (c) Seam and jump sheet

Figure 3.21: These sketches of a cross section through Ω show the three categorical possibilites for a contact change of maximal* balls associated with a segment between two points p and q in a contact domain $\mathcal{D}$. The contact domain is in a plane orthogonal to the viewer plane, such that $\overline{pq}$ lies in the viewer plane.

Algorithm 3 `searchcc`(p, q, $\mathcal{C}$) returns point on $\overline{pq}$ with different opposite contact

Precondition I: *C_p and C_q are not from the same contact class*

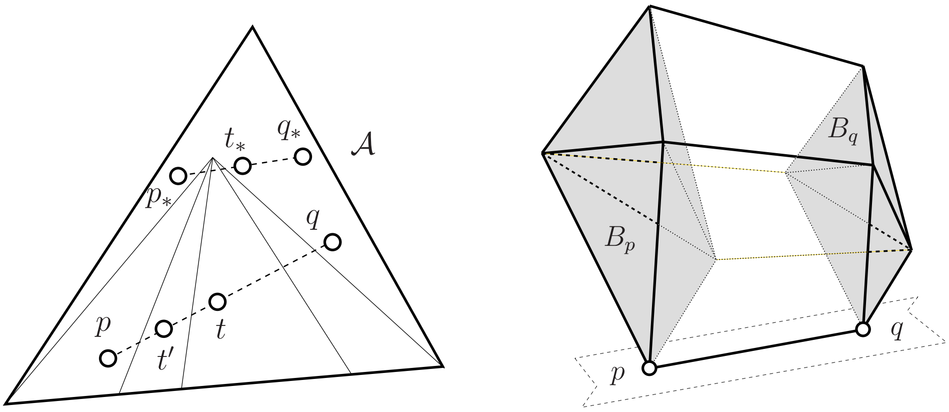
Precondition II: *C_p and C_q are not witnesses of a projection line crossing $\overline{pq}$*

```
procedure searchcc($p$, $q$, $\mathcal{C}$)
    $\mathcal{C}_p, \mathcal{C}_q \leftarrow$ opposite contact classes associated with $p, q$
    $t \leftarrow$ midpoint between $p$ and $q$
    $\mathcal{C}_t \leftarrow$ opposite contact class associated with $t$

    if $\mathcal{C}_t = \mathcal{C}_p$ then
        return searchcc($t$, $q$, $\mathcal{C}$)
    else if $\mathcal{C}_t = \mathcal{C}_q$ then
        return searchcc($p$, $t$, $\mathcal{C}$)
    else
        return $t$
    end if
end procedure
```

3.3.2.3.2 Equal opposite contact classes

Now assume that the maximal* balls associated with p and q have an opposite contact of the same contact class. In order to verify that no projection line crosses $\overline{pq}$, the family of maximal* balls projecting on this segment has to be contained in Ω. This family of maximal* balls spans a convex polyhedron, which is identical to the convex hull of all vertices on the boundary of B_p and B_q (see Figure 3.22b). A pseudo-code description of the according algorithm `sweep`, which takes $\mathcal{C}$ and the two maximal* balls B_p and B_q as arguments, is given in Algorithm 4. The most expensive step is the one at line 6, where the obtained convex hull and $\partial\Omega$ are checked for intersections.



(a) Binary search

(b) Sweep

Figure 3.22: (a) Binary search to find witnessing contact classes. The solid lines represent the final contact arrangement $\mathcal{A}$. (b) The convex hull resulting from the vertices on B_p and B_q.

Algorithm 4 `sweep`(B_p, B_q, $\mathcal{C}$) returns **true** if no intersection with $\partial\Omega$, **false** otherwise
Precondition: C_p *and* C_q *are from the same contact class*

```
procedure sweep($B_p$, $B_q$, $\mathcal{C}$)
    $\mathcal{C}_{pq}$ ← opposite contact class of $B_p$ and $B_q$
    $y$ ← shape component of $\mathcal{C}$
    $y'$ ← shape component of $\mathcal{C}_{pq}$
    $CH$ ← convex hull of vertices on $\partial B_p$ and $\partial B_q$

    if $CH \cap (\partial\Omega \setminus \{y, y'\}) = \emptyset$ then
        return true
    else
        return false
    end if
end procedure
```

If `sweep`(B_p, B_q, $\mathcal{C}$) detects an intersection, a point on $\overline{pq}$ with an associated opposite contact of a different contact class has to exist, and is found by the algorithm `searchcc*` (see Algorithm 5) that works similar to `searchcc`.

While `searchcc` starts with two points whose opposite contact classes are different, `searchcc*` does not. That makes repeated use of the `sweep`-algorithm (line 8) necessary. See again Figure 3.22a for an example. The midpoint t_* between p_* and q_* lies again in the same face of the final contact arrangement $\mathcal{A}$. Only by use of the `sweep` function, it can be decided if $\overline{p_* t_*}$ or $\overline{t_* q_*}$ is crossed by a projection line.

If no intersection is detected by `sweep`(B_p, B_q, $\mathcal{C}$), and thus the line segment $\overline{pq}$ is not crossed by any projection line, then this segment is called *crossing-free.* The final subdomains of the contact arrangement are characterized by the fact that they are bounded by crossing-free segments, and are then also called crossing-free. A proof of this property, valid for all shapes which do not contain inclusions, is given in Section 3.3.2.4

Algorithm 5 `searchcc*`(p, q, $\mathcal{C}$) returns point on $\overline{pq}$ with different opposite contact

Precondition I: C_p *and* C_q *are from the same contact class*

Precondition II: `sweep`(B_p, B_q, $\mathcal{C}$) = **false**

```
procedure searchcc*($p$, $q$, $\mathcal{C}$)
    $B_p \leftarrow$ maximal* ball projecting on $p$
    $\mathcal{C}_{pq} \leftarrow$ opposite contact class of $B_p$ and $B_q$
    $t \leftarrow$ midpoint between $p$ and $q$
    $\mathcal{C}_t \leftarrow$ opposite contact class associated with $t$
    $B_t \leftarrow$ maximal* disk projecting on $t$

    if $\mathcal{C}_t = \mathcal{C}_{pq}$ then
        if sweep($B_p$, $B_t$, $\mathcal{C}$) = true then
            return searchcc*($t$, $q$, $\mathcal{C}$)
        else
            return searchcc*($p$, $t$, $\mathcal{C}$)
        end if
    else
        return $t$
    end if
end procedure
```

3.3.2.3.3 Putting things together

Consider a contact class $\mathcal{C}$, with a fixed representative vertex $\tilde{v}$, and two points p and q in its contact domain. Let B_p and B_q be the two maximal* balls that project on p and q respectively, and $\mathcal{C}_p$ and $\mathcal{C}_q$ the accordant opposite contact classes. Then the procedure `projline`(p, q, $\mathcal{C}$), described in Algorithm 6, returns a projection line that crosses $\overline{pq}$ on the contact plane, if one exists, or $\varnothing$ if the segment is crossing-free.

If $\mathcal{C}_p = \mathcal{C}_q$, queried in line 4, and the `sweep` algorithm does not indicate an intersection with $\partial\Omega$ (line 5), then $\overline{pq}$ is crossing-free. If `sweep` detects an intersection, then crossing projection lines exist. This means that a point on $\overline{pq}$ with a different associated opposite contact class has to exist. This point is found with help of the `searchcc*` algorithm and a recursive call of `projline` is invoked (see lines 8 and 9).

If $\mathcal{C}_p \neq \mathcal{C}_q$ then at least one crossing projection line exists. If these two contact classes are witnesses to a projection line L (checked in line 12), the `projline` algorithm terminates by returning L. Otherwise, two points on $\overline{pq}$ whose opposite contact classes are witnessing a single projection line are found by applying binary search of `searchcc` and, again, recursive calls of `projline` as invoked in lines 16 and 17. For a flow chart representation of the outlined procedure consult Figure 3.23.

3.3.2.4 Arrangement computation: the algorithm

For any given point on a contact domain the associated maximal* ball can be constructed as described in Section 3.3.2.2. For two points on the domain it can be decided if there exists a projection line, induced by a pseudo-seam or a seam (possibly together with a jump edge) that crosses the connecting segment. If a crossing line exists, it can be found. As will be

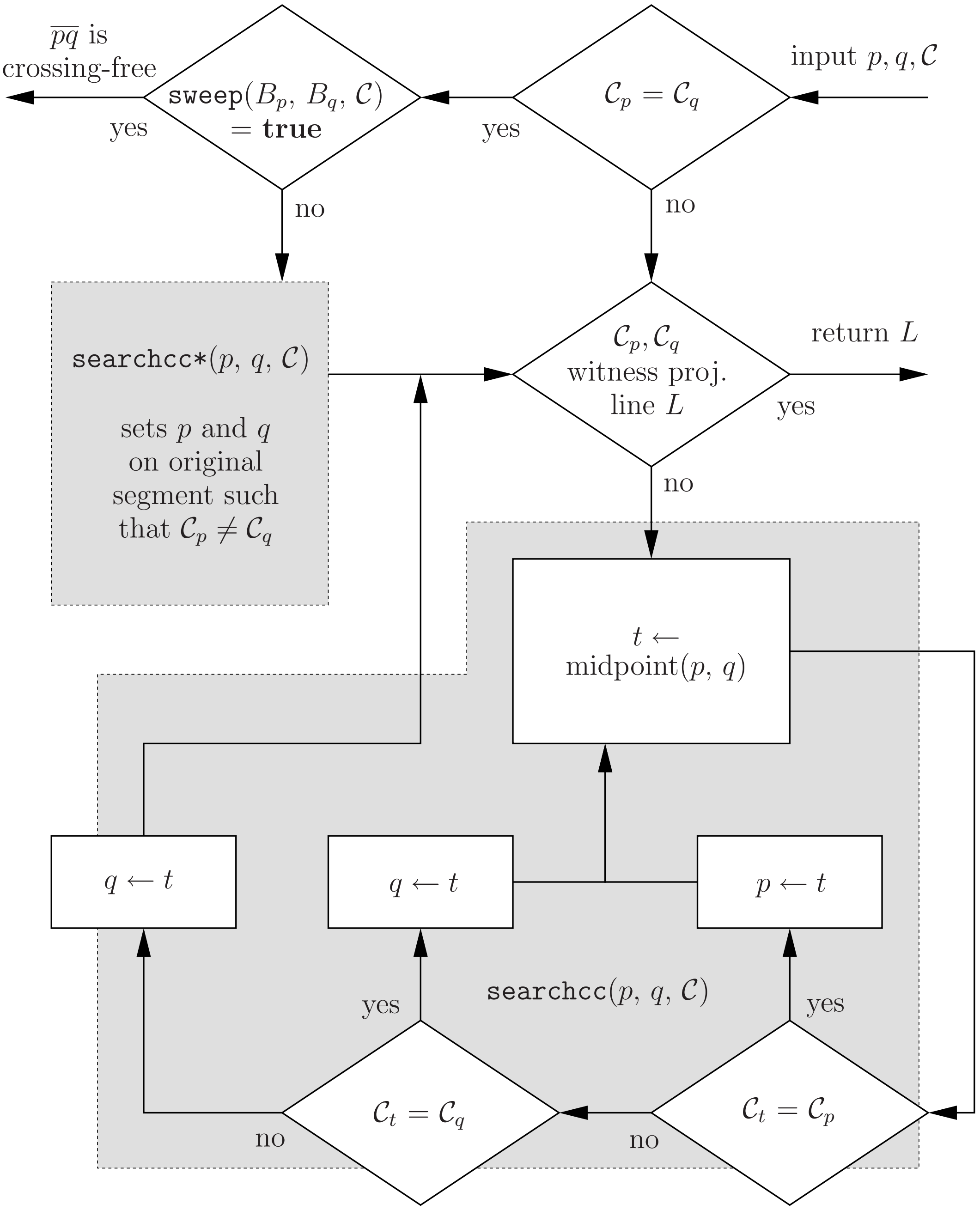

$\overline{pq}$ is crossing-free
yes
sweep(B_p, B_q, $\mathcal{C}$) = **true**
yes
$\mathcal{C}_p = \mathcal{C}_q$
input $p, q, \mathcal{C}$
no
no
searchcc*(p, q, $\mathcal{C}$)
sets p and q on original segment such that $\mathcal{C}_p \neq \mathcal{C}_q$
$\mathcal{C}_p, \mathcal{C}_q$ witness proj. line L
return L
yes
no
$t \leftarrow$ midpoint(p, q)
$q \leftarrow t$
$q \leftarrow t$
$p \leftarrow t$
yes
searchcc(p, q, $\mathcal{C}$)
yes
$\mathcal{C}_t = \mathcal{C}_q$
no
$\mathcal{C}_t = \mathcal{C}_p$
no

Figure 3.23: This flow chart summarizes the algorithm `projline` which finds a possible projection line, that crosses the segment $\overline{pq}$ in the contact domain of $\mathcal{C}$.

Algorithm 6 `projline`(p, q, $\mathcal{C}$) returns projection line that crosses $\overline{pq}$

```
procedure projline($p$, $q$, $\mathcal{C}$)
    $B_p, B_q \leftarrow$ maximal* balls projecting on $p, q$
    $\mathcal{C}_p, \mathcal{C}_q \leftarrow$ opposite contact classes of $B_p, B_q$

    if $\mathcal{C}_p = \mathcal{C}_q$ then                          ▷ equal opposite contact classes
        if sweep($B_p$, $B_q$, $\mathcal{C}$) = true then               ▷ check if $\overline{pq}$ is crossing-free
            return $\varnothing$
        else
            $t \leftarrow$ searchcc*($p$, $q$, $\mathcal{C}$)
            return projline($p$, $t$, $\mathcal{C}$)
        end if
    else                                                ▷ different opposite contact classes
        if $\mathcal{C}_p$ and $\mathcal{C}_q$ are witnesses to a projection line crossing $\overline{pq}$ then
            $L \leftarrow$ projection line witnessed by $\mathcal{C}_p, \mathcal{C}_q$
            return $L$
        else
            $t \leftarrow$ searchcc($p$, $q$, $\mathcal{C}$)
            return projline($p$, $t$, $\mathcal{C}$)
        end if
    end if
end procedure
```

shown in the following, this is all that is needed to build the contact arrangement in a given contact domain.

As said in Section 3.3.2.1, the basic idea is to split the initial contact domain into subdomains which are not intersected by any projection lines. To decide if a given subdomain D is crossing-free, and thus does not have to be divided any further, its bounding edges are inspected for crossings. The procedure `projline`(p, q, $\mathcal{C}$) requires the two associated maximal* balls B_p and B_q to have one unequivocal opposite contact, respectively. If one of the two

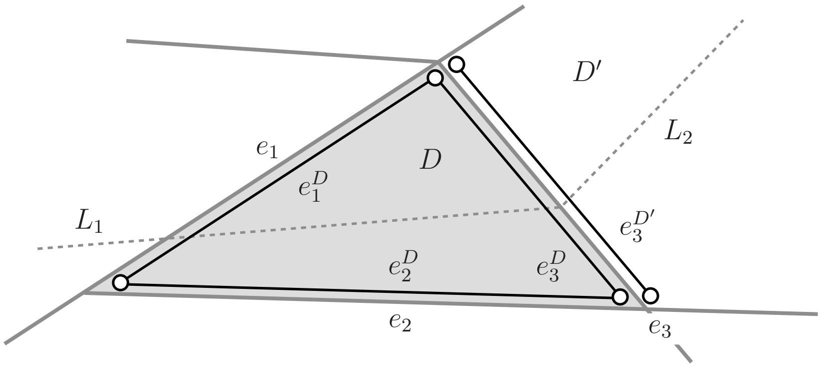


Figure 3.24: The projection line L_1 is detected by segment e_3^D, associated with D, while L_2 is detected by the opposite segment, associated with D'.

points is chosen on a projection line, which may partially define the boundary of a subdomain, then this criterion is not always satisfied. To avoid this, the segment e^D, associated with a boundary edge e of a subdomain D, is chosen such that

- $\mathrm{HD}(e, e^D) < \varepsilon$ for an $\varepsilon > 0$, and
- $e^D \subset D \setminus \partial D$.

In other words, the endpoints of e^D lie infinitesimally close to the endpoints of e, but still inside D. An example is given in Figure 3.24, with D being the subdomain of interest. The two dashed grey lines, representing projections of two different (pseudo-)seams, do not yet participate in the intermediate (solid grey) arrangement, but are part of the final contact arrangement. The segment e_2^D is obviously crossing-free, while e_1^D and e_3^D, when passed via their endpoints to the `projline` algorithm, both return L_1, being the supporting projection line of one projected (pseudo-)seam. Note however that the edge $e_3^{D'}$, associated with the neighboring subdomain D', returns a different projection line, namely L_2.

Property 3. *Let D be a subdomain bounded by the edges e_i. If all segments e_i^D are crossing-free, then D is crossing-free if $\partial\Omega$ is connected (or Ω does not contain inclusions).*

Proof. Assume that the aforementioned segments e_i^D are crossing-free with the shared opposite contact class $\mathcal{C}'$, but D is not crossing-free. This means that D contains a set of edges of the final contact arrangement, which can be separated from ∂D by a closed curve h. Let $\mathcal{H} \subset \mathcal{B}(\mathcal{C})$ be the set of maximal* balls that project on h. The opposite contact of every ball of $\mathcal{H}$ is of the class $\mathcal{C}'$, and their projections define another closed curve h' on the contact domain of $\mathcal{C}'$. There must exist a point on the bounded side of h on the contact plane of $\mathcal{C}$, which has an associated opposite contact of a class $\mathcal{C}^* \neq \mathcal{C}'$. All maximal* balls of $\mathcal{H}$, as well as the two domains bounded by h and h' in their respective contact domain, are clearly free of intersections with $\partial\Omega$ that are not induced by $\mathcal{C}$ or $\mathcal{C}'$. Thus the part of $\partial\Omega$ inducing a contact of type $\mathcal{C}^*$ must lie *inside* the convex hull spanned by the set $\mathcal{H}$. As a consequence, $\partial\Omega$ has to be disconnected. □

Given a triangulated solid without inclusions, the final observation in Property 3 now allows the formulation of the algorithm `arrang(`$\mathcal{C}$`)` which computes the contact arrangement $\mathcal{A}(\tilde{x}, y)$ for a regular typed contact class $\mathcal{C} = (\tilde{x}, y)$. See Algorithm 7 for a pseudo-code description of the procedure, which is now discussed in more detail.

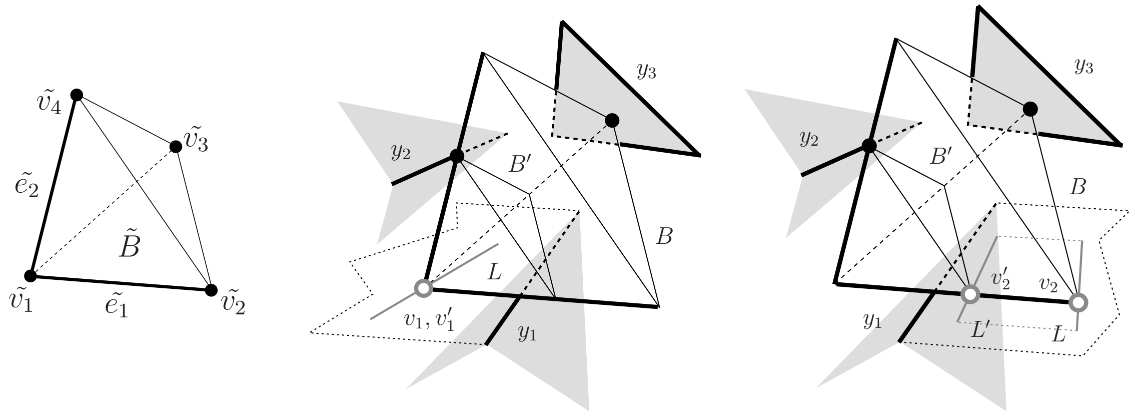
The procedure takes as input a contact class $\mathcal{C}$ of regular type. In the lines 2 to 7 several necessary initializations are invoked: a representative vertex, and to compute the contact domain $\mathcal{D}$ with respect to it is computed. The boundary of $\mathcal{D}$ is already part of the contact arrangement $\mathcal{A}$.

The first major subpart of `arrang` is responsible for the decomposition of the contact domain into crossing-free subdomains (lines 8 to 24). Due to Property 3 it is possible to determine via the boundary edges of a subdomain (in the first step the whole contact domain) if it is crossing-free (see line 13). If one of these edges is not crossing-free (detected in line 19), it induces at least one projection line L which splits the subdomain into two new subdomains. The above is applied recursively on these new subdomains, until only crossing-free ones are left. Note again: all points of a crossing-free subdomain are projections of maximal* balls featuring an opposite contact of the same contact class, and are thus centered on the same sheet of $\mathrm{MA}_{\tilde{B}}(\Omega)$.

Algorithm 7 `arrang`($\mathcal{C}$) returns the final contact arrangment for the contact class $\mathcal{C}$

```
procedure arrang($\mathcal{C}$)
    $\tilde{v} \leftarrow$ representative vertex of the ball component of $\mathcal{C}$
    $\mathcal{D} \leftarrow$ contact domain of $\mathcal{C}$ with respect to $\tilde{v}$
    $\mathcal{A} \leftarrow$ empty arrangement
    $Q \leftarrow$ empty stack of subdomains
    $Q$.push($\mathcal{D}$)
    $\mathcal{A}$.insert($\partial\mathcal{D}$)

    while $Q$ is not empty do                    ▷ decompose into crossing-free subdomains
        $D \leftarrow Q$.pop()
        $k \leftarrow$ number of boundary segments of $D$
        $i \leftarrow 1$
        $L \leftarrow \varnothing$
        while $i \leq k \wedge L = \varnothing$ do
            $e \leftarrow$ i$^{th}$ segment of $\partial D$
            $i \leftarrow i + 1$
            $p, q \leftarrow$ endpoints of $e^D$
            $L \leftarrow$ projline($p$, $q$, $\mathcal{C}$)
        end while
        if $L \neq \varnothing$ then
            $D_1, D_2 \leftarrow$ decomposition of $D$ via projection line $L$
            $Q$.push($D_1$, $D_2$)
            $\mathcal{A}$.insert($D \cap L$)
        end if
    end while
    for all edges $a$ of $\mathcal{A}$ do          ▷ remove dispensable arrangement edges
        $D_1^a, D_2^a \leftarrow$ the two subdomains incident to $a$
        $\mathcal{C}_1^a, \mathcal{C}_2^a \leftarrow$ the two opposite contact classes associated with $D_1^a, D_2^a$
        if $\mathcal{C}_1^a = \mathcal{C}_2^a$ then
            $\mathcal{A}$.remove($a$)
        end if
    end for
    return $\mathcal{A}$
end procedure
```



(a) Unit ball (b) Projection to line (c) Projection to region

Figure 3.25: Depending on the chosen representative vertex, a jump sheet may project to a one- or a two-dimensional component in the contact domain.

All edges shared by two crossing-free subdomains are now part of the intermediate arrangement $\mathcal{A}$. In the second subpart of `arrang` (lines 25 to 31), artifact edges between neighboring subdomains that describe the same sheet are removed from $\mathcal{A}$. This results in the final contact arrangement.

A convex region that is intersected by a line, is by this line always split into two regions that are again convex. The contact domains for a contact class of type *EE* and *FV* however, are not necessarily convex. This has the unwanted effect that a projection line may split a domain into more than two subdomains. To prevent this, a non-convex contact domain is initially split by dummy lines into simple convex subdomains which are used to initialize the stack in the `arrang` algorithm. These dummy lines are removed in the second subpart of `arrang`. In the final arrangement non-convex subdomains of course may be created due to deleting artifact edges.

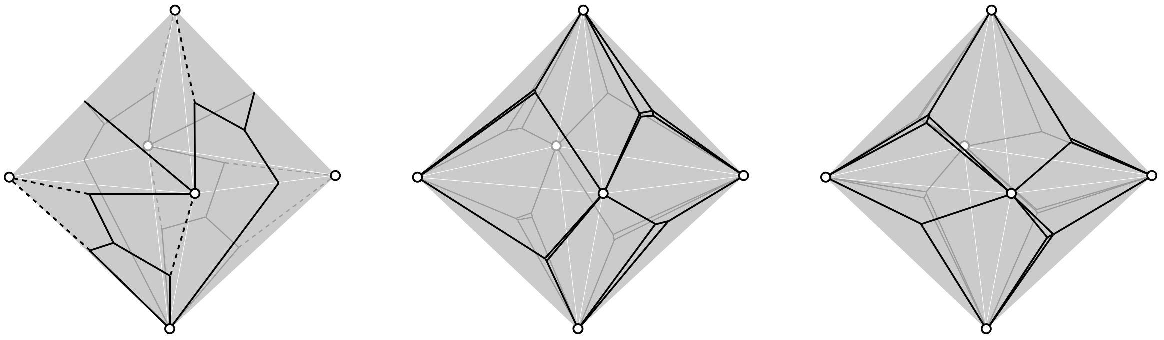
Vertex-dependent jump projections

For a contact class of type *EE* the representative vertex is one of two possible vertices on the unit ball component, for a contact class of type *FV* there are even three possible choices. As mentioned in Section 3.3.1.2 and Section 3.3.2.3.1, a jump sheet of the medial axis may, depending on this chosen representative vertex, project to a one-dimensional component on the contact plane, while for another representative vertex the jump sheet may correspond to a two-dimensional component (i.e., face) of the contact arrangement. Consult Figure 3.25 for an example where the contact domains (shown dotted black) are associated with the contact class $\mathcal{C} = (\tilde{e}_1, y_1)$ of type *EE* with respect to two different representative vertices $\tilde{v}_1$ in (b) and $\tilde{v}_2$ in (c). The maximal* ball B is centered on the seam s defined by $\mathcal{B}(\mathcal{C}, (\tilde{e}_2, y_2), (\tilde{v}_3, y_3))$, while B' is centered on the pseudo-seam s' defined by $\mathcal{B}(\mathcal{C}, (\tilde{v}_4, y_2))$. The two contact classes $\mathcal{C}$ and $(\tilde{e}_2, y_2)$ are neighboring on $\partial\tilde{B}$ and so induce a jump sheet ς.

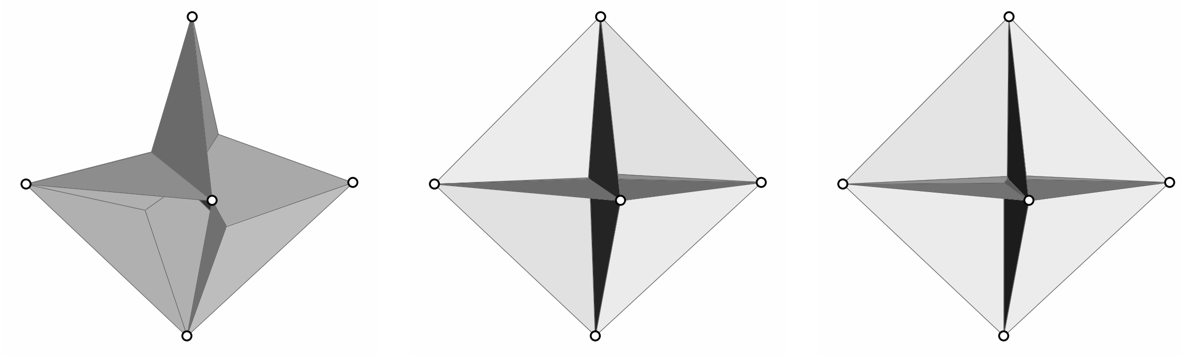
(b): Let $\tilde{v}_1$ be the representative vertex. It lies in the intersection of the two ball components $\tilde{e}_1$ and $\tilde{e}_2$ which participate in the contact classes generating the jump sheet. As a consequence the projections of s and s' in the contact domain with respect to $\tilde{v}_1$ coincide at the projection line L. The jump sheet ς is spanned by s and s' and as such also projects to L.

(c): For $\tilde{v}_2$ being the representative vertex, s and s' project to two different projection lines L and L' respectively. The region between L and L' on the accordant contact domain is part of the two-dimensional projection of the jump sheet ς.

$\tilde{B}$ with 4 facets $\tilde{B}$ with 20 facets $\tilde{B}$ with 128 facets



contact arrangements



medial axes

Figure 3.26: Contact arrangements (top row) and medial axes (bottom row) of a slightly perturbed octahedron with respect to polyhedral unit balls with 4, 20 and 128 facets (from left to right). Dashed lines are projections of pseudo-seams.

3.3.3 Assembling the medial axis and offset

Once the contact arrangements of all regular typed contact classes, that occur for a unit ball $\tilde{B}$ and a triangulated solid Ω, have been computed, the medial axis components as well as their interconnections can be assembled easily. Based on the obtained data representation, the problem of trimmed offset computation is addressed.

3.3.3.1 Assembling the medial axis from its projections

As stated in detail in Section 3.3.1.2, every sheet can be associated with two faces on two different contact arrangements, every seam with three edges on three different contact arrangements, and so on. By this means every component of the medial axis is represented by at least one component of a contact arrangement. So the assembling of the medial axis can be performed by a simple computation as follows.

The medial axis $\mathrm{MA}_{\tilde{B}}(\Omega)$, with $\tilde{B}$ and Ω being triangulated solids, can be represented as a triangulated non-manifold mesh. In the non-degenerate case, such a mesh is completely described by its triangles. To store also connectivity information, the mesh can be embedded into a more sophisticated data-structure, as e.g. the *radial edge structure* introduced in [84].

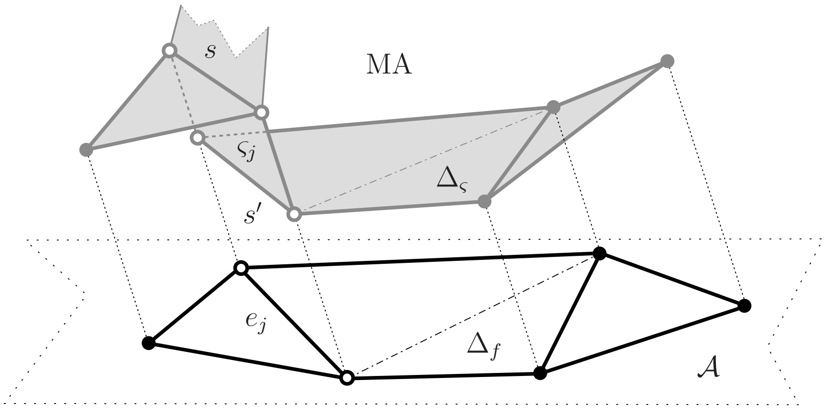


Figure 3.27: Part of a contact arrangement $\mathcal{A}$. The center points of the associated maximal* balls span a part of the medial axis MA.

In an arbitrary contact arrangement $\mathcal{A}$ consider a face f that has no segment on its boundary which results from a jump projection. Then for every vertex on the boundary of f, there exists an unequivocal maximal* ball that projects on it and that is known. Let ς be the sheet of the axis whose projection in $\mathcal{A}$ is f. Then every triangle Δ_f of the polygon triangulation of f is the projection of a triangle Δ_ς of the polygon triangulation of ς. Moreover, Δ_ς is spanned by the center points of the three maximal* balls that are associated with the three vertices of Δ_f.

If a boundary edge e of a triangle Δ_f is the projection of a jump sheet, then special care has to be taken. In general every point on such a jump projection edge is the projection of a whole family of maximal* balls, where the largest ball of this family is centered on a seam s, and the smallest one on a pseudo-seam s' (as discussed in Section 3.3.2.4). In practice, this means that an endpoint of e is associated with two maximal* balls, $B(s)$ and $B(s')$. If one of the contacts featured by $B(s)$ is of the opposite contact class associated with f, then the center point of $B(s)$ has to be chosen to span the according Δ_ς, otherwise the center of $B(s')$. Note, that also the jump sheet part represented by the jump projection edge e has to be dealt with. If s and s' meet in the center point of a maximal* ball that projects to an endpoint of e, then the part of the jump sheet that is associated with e is a triangle, otherwise a quadrilateral which is split into two triangles.

By iterating over the faces, edges and vertices of all contact arrangements, all medial axis components–spanned by the center points of the maximal* balls that project on the arrangement vertices–can be computed. See Figure 3.27 for an illustrative example, where the edge e_j denotes a jump projection edge, which is the projection of all maximal* balls centered on the seam s, the jump sheet ς_j and the pseudo-seam s'. Polygon triangulations of arrangement facets are adopted by the associated axis sheets: Δ_f is the projection of Δ_ς.

As most components project into more than one contact arrangement, duplicates have to be avoided when iterating over all arrangements. This is done by sorting out the primitives via the set of defining contact classes, which is unique for every component. As an example consider a slightly perturbed octahedron Ω and its medial axis with respect to several polyhedral unit balls $\tilde{B}$, where the number of facets increases from 4 to 128. The results are shown in Figure 3.26. Since Ω is convex in this example, all contact domains are contained in the facets of Ω and only contact classes of type *VF* occur. Consequently, the projections and contact arrangements can be visualized directly on $\partial\Omega$ (shown in the first row). The medial axis of the octahedron Ω with respect to the Euclidean unit ball consists of three squares which intersect each other along their diagonals. The medial axis with respect to a sparse polyhedral unit ball (a tetrahedron) is quite different (bottom left), since some of the vertices of the ball fit into the edge and vertex wedges of of the domain. When using a polyhedral unit ball with a larger number of facets (bottom center and right), however, the structure of the computed medial axis gets (as expected) more similar to the Euclidean case.

Connectivity

Basically all connectivity information can be read directly from the contact arrangements. A sheet is incident to a seam or pseudo-seam, if their projections are incident on any contact arrangement. A seam or pseudo-seam is incident to a (pseudo-)junction, if their projections are.

Again, things get a bit more complex for a jump projection edge e in a contact arrangement $\mathcal{A}$, as it represents a pseudo-seam s', a seam s and (the part of) a jump sheet at the same time. Obviously, the jump sheet is incident to s' and s. Consider a sheet ς that projects to a face f in $\mathcal{A}$ that is incident to e. If one of the defining contact classes of s is equal to the opposite contact class associated with f, then ς is incident to s, otherwise to s'. In a similar fashion the connectivity information between a jump projection edge and its two endpoints is obtained.

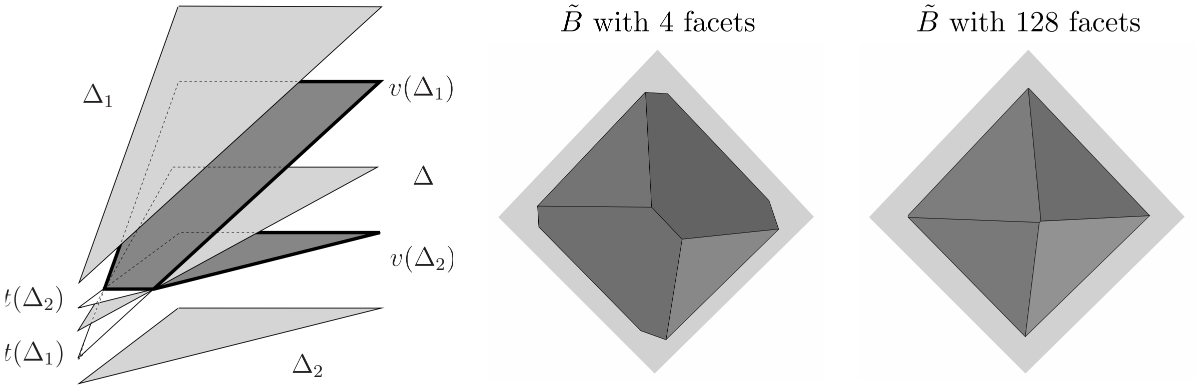
3.3.3.2 Assembling the offset from the medial axis

The medial axis is a useful tool for trimmed offset computation as already demonstrated in Section 2.4. While this is generally well-established in the two-dimensional case [1, 23], the structure has not yet been used much in 3-space for this purpose [15].

The medial axis representation as generated by the aforementioned algorithm is directly useful for offset computation with respect to the underlying linear quasi-metric. The sheets are the only axis components needed in the following. Each sheet ς of the medial axis is associated with two contact classes $\mathcal{C}_1$ and $\mathcal{C}_2$. A maximal* ball B with center point c_B on ς and scaling factor β has a unique point of contact p_i induced by $\mathcal{C}_i$ for $i \in \{1, 2\}$. Let ϱ be the offset size. Then the offset operation with ϱ applied to B gives us a new point p_i^ϱ for each of the two contact classes. This new point p_i^ϱ lies on the line defined by p_i and c_B. The position of p_i^ϱ with respect to the sheet ς determines whether or not it has to be trimmed:

- If $\beta > \varrho$ then p_i^ϱ lies between p_i and c_B. Therefore p_i^ϱ is a valid point of the offset surfaces.
- If $\beta < \varrho$ then c_B lies between p_i and p_i^ϱ. Therefore p_i^ϱ has to be trimmed.
- If $\beta = \varrho$ then $p_1^\varrho = p_2^\varrho = c_B$ and the point lies on the axis sheet where the trimmed and valid part of the offset surfaces join.

As mentioned above, the sheets of the piece-wise linear medial axis are stored as polygon triangulations. This results in one recurring configuration for the whole offset computation as visualized in Figure 3.28a. The three maximal* balls at the corner points are known.



(a) Trimming

(b) Trimmed offsets

Figure 3.28: (*a*) Basic configuration for offset computation. (*b*) Offsets of a slightly perturbed octahedron induced by unit balls with 4 and 128 facets.

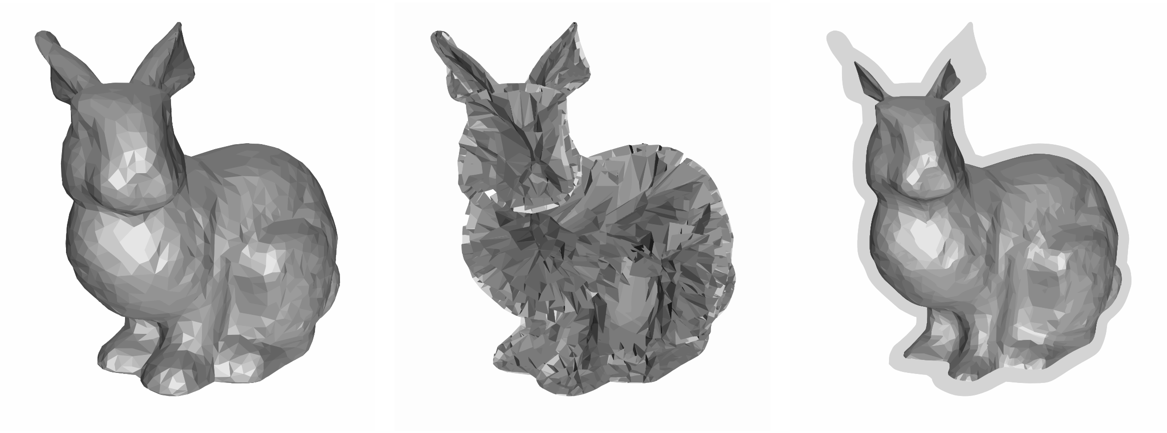
Depending on the offset size ϱ, certain parts of the triangles that lie on planes parallel to the contact planes define the valid offset surface. In particular, the triangle Δ of an axis sheet projects to two triangles Δ_1 and Δ_2 on two different contact arrangements. The offset surface generated from each of these triangles is split into a valid ($v(\Delta)$) and a trimmed ($t(\Delta)$) part, which intersect in the corresponding axis sheet. Note that a part derived from an *EE*- or *FV*-typed contact class resides on a plane which is partially defined by features of the unit ball.

Observation 13. *It has to be noted that the offset computed by the above procedure does not fully match with the one defined in Definition 1.2.7. Given a polyhedral unit ball $\tilde{B}$, let $\tilde{B}^*$ be the polyhedron that is centrally symmetric to $\tilde{B}$ with O being the center of symmetry. Then the offset we obtain from the medial axis $\mathrm{MA}_{\tilde{B}}(\Omega)$ is the trimmed inner offset of Ω with respect to $\tilde{B}^*$.*[5]

In Figure 3.28b the inner trimmed offset of an octahedron with respect to two different distance functions $\delta_{\tilde{B}}$ is shown.

---

[5] For a polyhedral unit ball $\tilde{B}$ which is centrally symmetric to itself, the notions coincide. The respective distance function $\delta_{\tilde{B}}$ then induces not only a quasi-metric, but a metric.



(a) Triangulated solid (b) Medial axis (c) Trimmed offset

Figure 3.29: The medial axis and trimmed offset of a 5000-facet instance of the Stanford bunny model with respect to a polyhedral unit ball with 128 facets.

3.3.4 Implementation and Examples

In the course of our work on medial axis computation with respect to piece-wise linear quasi-metrics, we implemented parts of the algorithm. Finally we assembled the single pieces to a program, which is now able to compute the medial axis (Figure 3.29b) as well as the trimmed inner offset (Figure 3.29c) of a triangulated solid (Figure 3.29a), with respect to a quasi-metric induced by an arbitrary convex polyhedron. The software is not ready for release, but basically fully functional.

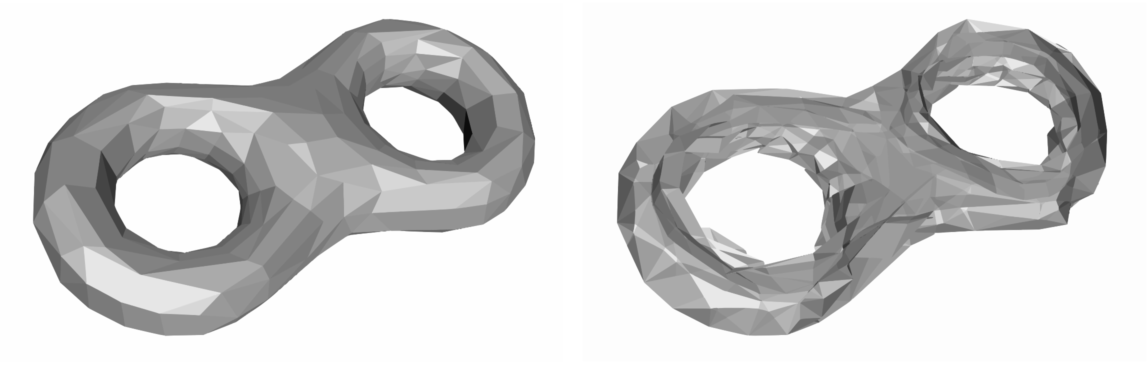
In Section 3.3.4.1 we will provide some details of the implementation done for the medial axis and offset computation, in particular regarding the underlying geometric library CGAL (Computational Geometry Algorithms Library) [90]. In Section 3.3.4.2 we will present some examples of different meshes and varying polyhedral unit balls, with runtimes and other statistical data of interest.

3.3.4.1 CGAL implementation details

The implementation has been done in the object-oriented programming language of C++. There have been several reasons for this choice:

- C++ is a widespread, popular and efficient programming language.
- The data representation and interaction between the shape and the unit ball recommends itself for an object-oriented approach.
- There exists an extensive C++ package for geometric problems, which provides data structures, algorithms for popular problems and robust implementations of exact number types. This collection of libraries is called CGAL [90].

Especially the prefabricated data structures of CGAL proved to be useful for the implementation. The `Polyhedron_3` structure, for example, is used to represent the shape–a



(a) Shape with genus 2 (b) Medial axis for tetrahedral $\tilde{B}$

Figure 3.30: Also the representation of shapes with genus > 0 is possible, as is the medial axis computation for such a shape.

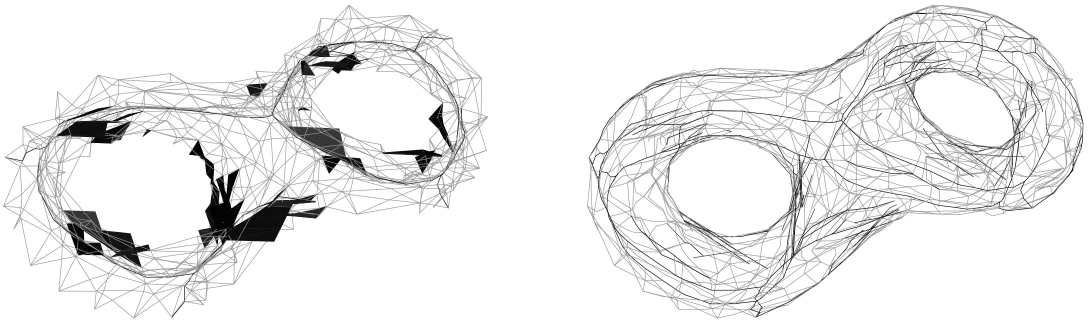
triangulated solid–as well as the unit ball via their boundary. Some details for the application, as given in [71], are the following:

> The polyhedral surface is realized as a container class that manages vertices, halfedges, facets with their incidences, and that maintains the combinatorial integrity of them. It is based on the highly flexible design of the halfedge data structure
> . . .
> Vertices represent points in space. Edges are straight line segments between two endpoints. Facets are planar polygons without holes. Facets are defined by the circular sequence of halfedges along their boundary. The polyhedral surface itself can have holes (with at least two facets surrounding it since a single facet cannot have a hole). The halfedges along the boundary of a hole are called border halfedges and have no incident facet. An edge is a border edge if one of its halfedges is a border halfedge. A surface is closed if it contains no border halfedges. A closed surface is a boundary representation for polyhedra in three dimensions.
> . . .
> Other intersections besides the incidence relation are not allowed.

One of the big advantages of this data structure, beside its use of the powerful halfedge data structure [72], and that it allows also the handling of shapes with genus > 0 (see Figure 3.30), is the possibility to embed its components in an `AABB_tree` [9], which facilitates intersection detection, as described in Section 3.3.2.2.1.

The contact arrangements of the regular typed contact classes are realized by instances of the `Arrangement_2` class, described in [53]:

> Given a set C of planar curves, the arrangement A(C) is the subdivision of the plane into zero-dimensional, one-dimensional and two-dimensional cells, called vertices, edges and faces, respectively induced by the curves in C.
> . . .
> This graph can be represented using a doubly-connected edge list data-structure (Dcel for short), which consists of containers of vertices, edges and faces and maintains the incidence relations among these objects. The main idea behind the Dcel data-structure is to represent each edge using a pair of directed halfedges



(a) Medial axis structure

(b) Contact arrangements

Figure 3.31: (*a*) Seams and jump sheets (black) and pseudo-seams (grey). (*b*) The respective projections define the contact arrangements.

> ...
> Halfedges are used to separate faces, and to connect vertices
> ...
> Each halfedge e stores a pointer to its incident face, which is the face lying to its left. Moreover, every halfedge is followed by another halfedge sharing the same incident face, such that the target vertex of the halfedge is the same as the source vertex of the next halfedge. The halfedges are therefore connected in circular lists, and form chains, such that all edges of a chain are incident to the same face and wind along its boundary.

In Figure 3.31 the medial axis components which project to arrangement segments (seams, pseudo-seams and jump sheets) are shown to the left, and their projections to the right. Note that contact arrangements for contact classes of type EE and FV do not lie on the boundary of Ω.

Another major contribution of the CGAL framework is that data structures can be initialized with user-defined number types. This is done with help of the *template*-mechanism, supported by the recent C++ architecture standards. For my implementation we chose the exact rational number type `Gmpq` [91]. As shown in Section 3.3.2.2, for $\partial\Omega$ and $\partial\tilde{B}$ given with rational vertex coordinates, all points obtained by the construction of a maximal* ball also have rational coordinates. As a consequence also the medial axis–spanned by rational center points of maximal* balls–is exactly representable by rational numbers as implemented by `Gmpq`.

For the two-dimensional representation of the final contact arrangements we use the CGAL-supported `Qt` library [89]. All 3D-objects are represented as *OOGL* objects, supported by the mesh-visualization program `Geomview` [68]. This is sufficient, as the medial axis as well as the trimmed offset surfaces are visually representable as triangle-meshes.

3.3.4.2 Runtime examples

The time needed for the computation of the contact arrangements depends on various criteria:

# mesh facets	**axis computation**	**offset computation**	**# axis sheets**
115	115.54 secs	1.87 secs	417
267	284.04 secs	4.15 secs	1021
575	525.19 secs	7.73 secs	2088
1396	1209.51 secs	15.63 secs	4653

Table 3.2: Computation times for the medial axis and trimmed offset with respect to a tetrahedral unit ball of various instances of the Venus model as depicted in Figure 3.33.

- The quality of the boundary mesh influences the time gain provided by the AABB-tree structure [9]. A mesh with roughly equally sized and equilateral triangles is advantageous, while single large triangles can impair the nesting balance of the tree.
- A rather complex and strongly branched shape has more reflex features thus inducing more contact classes of type EE and FV. This means that more contact arrangements have to be computed, and also the construction of the contact domain is more costly for these contact types.
- The nesting complexity of a single contact arrangement, however, which also has a certain influence on the performance, is in average higher for less ramified shapes, as in general more components of the medial axis project on the accordant contact domain.
- Besides the combinatorial sizes of the triangulated solid and the unit ball, also the mutual geometric alignment of them plays a role when it comes to the complexity of the resulting axis (remember the worst case example in the planar case as depicted in Figure 3.11). A complex axis means high computation costs.

For all these reasons it is hard to provide theoretical bounds for the algorithms complexity and runtime, which is why in the following we will concentrate on empirical data obtained from test runs on several instances of mesh models with varying properties.

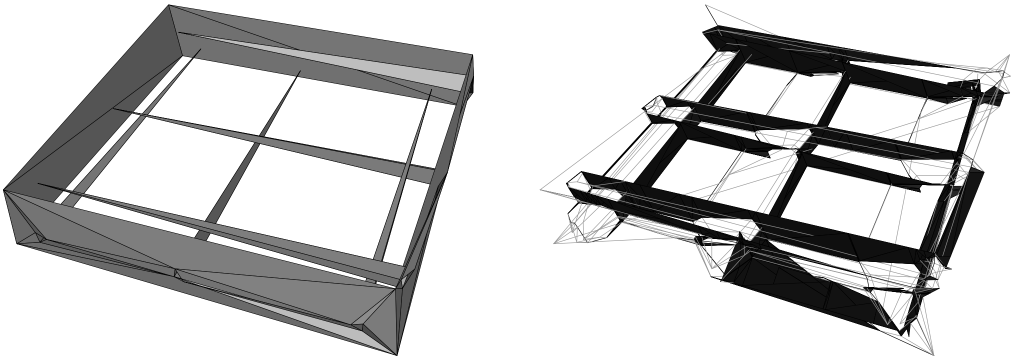
As a first example we consider the simple “Venus” model as it has also been used in [33]. In Figure 3.33 several mesh instances of this shape are shown, together with different representations of the medial axis with respect to a unit ball with 4 facets. The respective computation times compare favorably with the ones reported in [33] for the same shape.[6] There, the computation of the medial axis for the Venus shape with 250 facets is performed in 5.6 hours, with computing times growing considerably with respect to the number of faces. As can be seen in Table 3.2, the new algorithm computes the exact medial axis with respect to the quasi-metric induced by a tetrahedral $\tilde{B}$ for an instance of comparable size (267 facets) in less than 5 minutes. Also, the approach presented in [33] is likely to suffer from much worse runtime development for larger input sizes, than the algorithm introduced herein. See again Table 3.2, which indicates an almost linear relation between mesh size and computation time. Computation times for the trimmed offset are linear to the combinatorial axis size, and as such very moderate.

---

[6] This is the only implementation we are aware of that constructs the exact medial axis with respect to a specific metric. According to [33] some effort has been put into optimzation, but still runtimes have to be regarded with care, as they are obtained from an older SGI system, running with 300 MHz.

# mesh facets	axis computation	offset computation	# axis sheets
174	165.10 secs	2.29 secs	559
404	473.47 secs	8.84 secs	1657
860	923.99 secs	16.94 secs	3590
1814	1922.14 secs	35.95 secs	7736
3490	3926.88 secs	69.23 secs	14936

Table 3.3: Axis and offset computation times for different instances of the Lion model.



(a) Iron maiden pizza box

(b) Medial axis seam structure

Figure 3.32: Iron maiden pizza box mesh (with omitted top and bottom facets), and its medial axis seam and jump sheet structure (with respect to a tetrahedral unit ball).

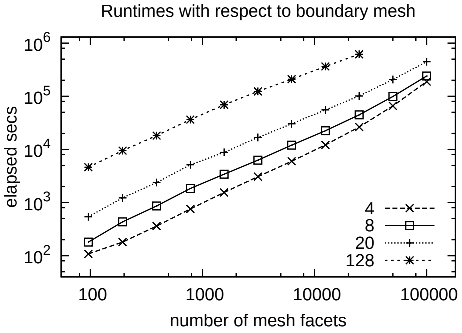
The Lion model (some mesh instances together with medial axis and trimmed offsets are depicted in Figure 3.34) represents a slightly more ramified shape. However, like for the Venus model, the axis size as well as the time needed for its computation, grows in an almost linear fashion with respect to the input mesh size.

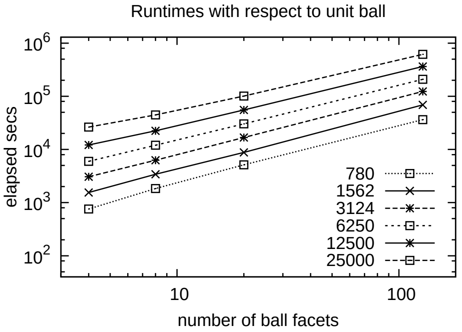
The more convoluted “Iron-maiden pizza box” mesh (also introduced in [33]), has 60 facets, but already a medial axis with 583 sheets, when computed with respect to a tetrahedral unit ball (see Figure 3.32). This is clearly more than for the Venus mesh with 115 and the Lion mesh with 174 facets.

The most extensive tests have been performed with the Armadillo model. It comes in mesh instances of combinatorial sizes between 100 and 100000 facets, and is as such the most diversified among the models we considered. With help of these mesh-instances (see Figure 3.35 for some examples), experimental results that indicate the relation between the complexity of the input mesh data (number of facest on $\partial\Omega$ and $\partial\tilde{B}$), the computing times and the size of the generated output have been obtained. The computation times for the Armadillo example, reported in Table 3.4, grow only slightly super-linearly with respect to the number of facets of the mesh, and even sub-linearly with respect to the number of facets in the unit ball. In addition, when analyzing the relation between the size (i.e., the number of planar sheets) of the computed medial axis and the number of facets on the boundaries of Ω and of $\tilde{B}$, it comes into notice, the size of the medial axis grows linearly with the size of $\partial\Omega$, but only very slowly (much less than linear) with the size of ∂B. See Table 3.5.

# facets	4	8	20	128
96	1.08E+2	1.80E+2	5.40E+2	4.61E+3
194	1.80E+2	4.32E+2	1.22E+3	9.47E+3
390	3.60E+2	8.64E+2	2.38E+3	1.82E+4
780	7.56E+2	1.84E+3	5.15E+3	3.65E+4
1562	1.55E+3	3.42E+3	8.78E+3	6.90E+4
3124	3.06E+3	6.26E+3	1.68E+4	1.23E+5
6250	5.98E+3	1.20E+4	3.02E+4	2.09E+5
12500	1.21E+4	2.24E+4	5.54E+4	3.64E+5
25000	2.64E+4	4.46E+4	1.01E+5	6.14E+5
50000	6.58E+4	9.89E+4	2.07E+5	–
100000	1.88E+5	2.40E+5	4.47E+5	–

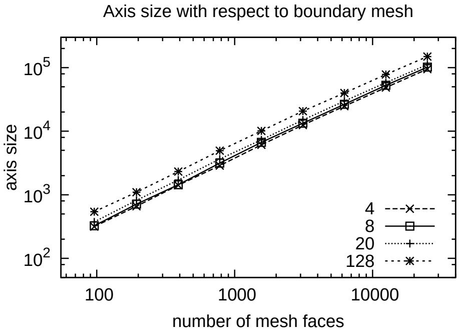
Table 3.4: Left: Computation times (in seconds) for several polyhedral unit balls (shown in the different columns; the first row specifies the number of faces) and various instances of the Armadillo mesh (shown in the rows) on a single CPU with 2.5 GHz. Right: Results plotted on a log-log scale.

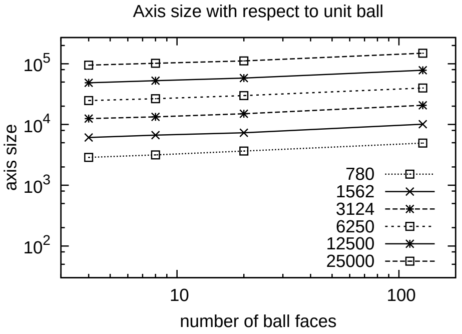




# facets	4	8	20	128
96	315	325	375	542
194	661	714	819	1097
390	1410	1437	1709	2315
780	2879	3154	3661	4945
1562	6106	6689	7316	10091
3124	12514	13365	15043	20655
6250	24764	26519	29841	39906
12500	48592	52655	58055	78155
25000	94715	101733	111355	148967

Table 3.5: Number of sheets of the medial axis for several polyhedral unit balls (shown in the different columns; the first row specifies the number of faces) and various instances of the Armadillo mesh (shown in the rows). Right: Results plotted on a log-log scale.





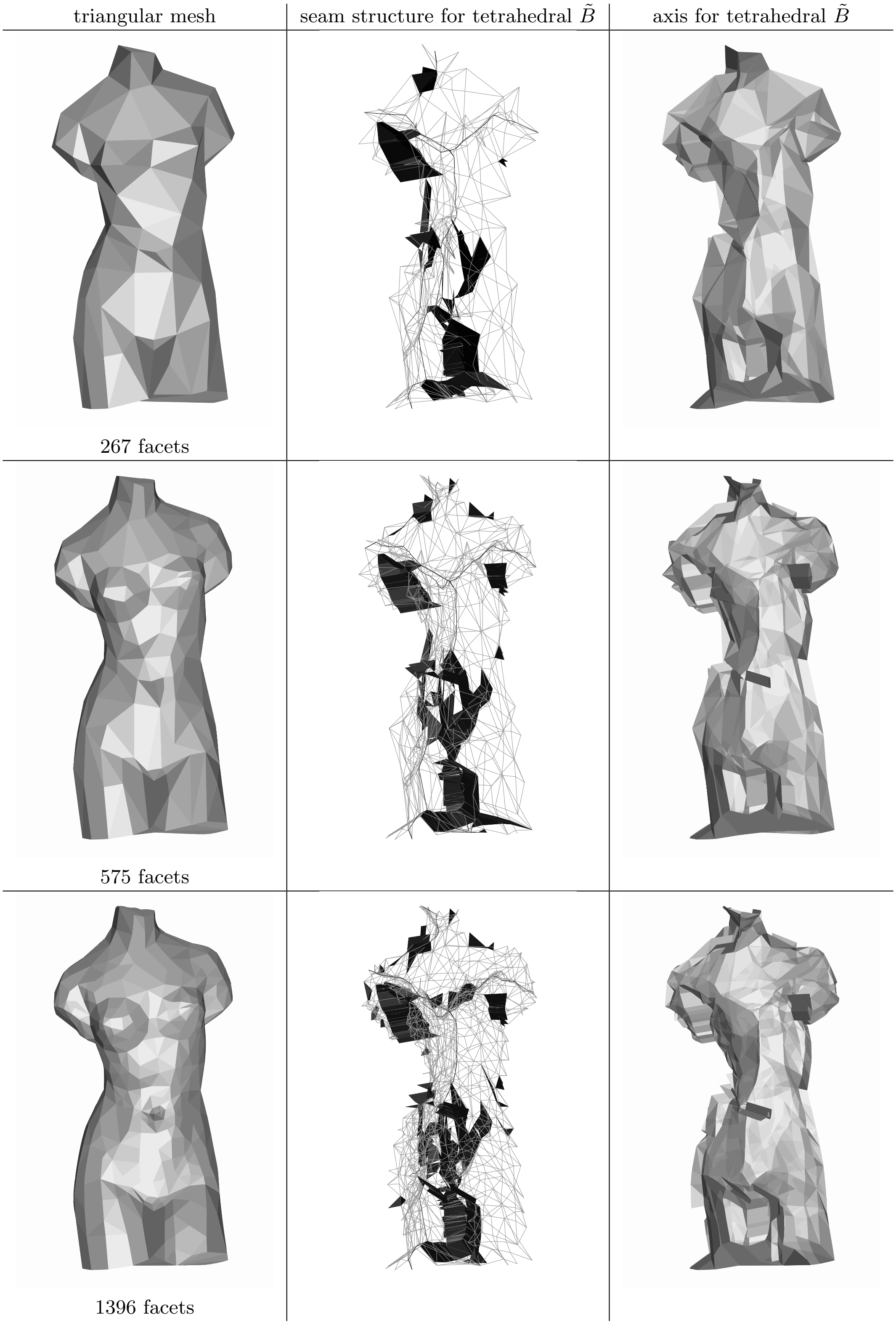
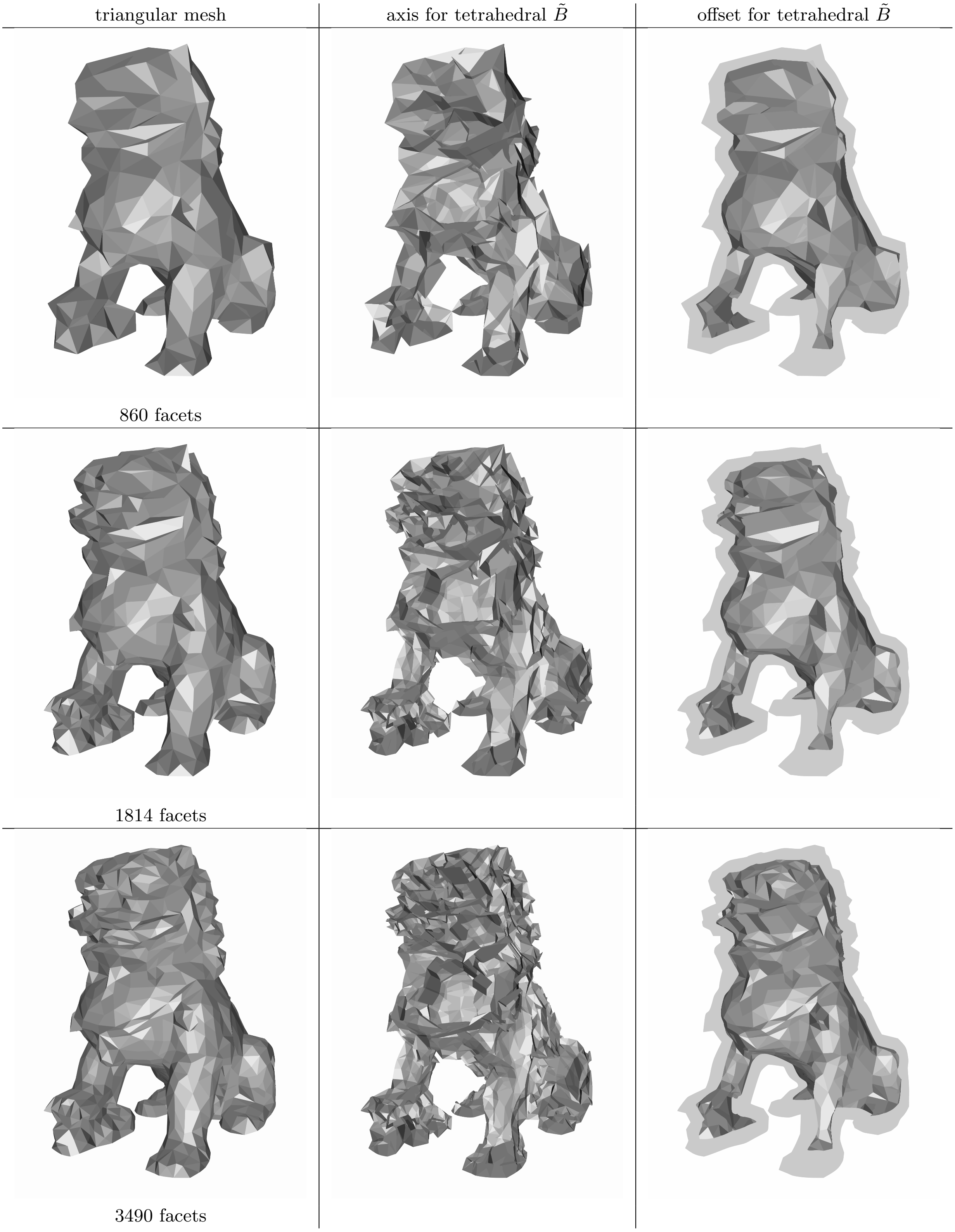

triangular mesh
seam structure for tetrahedral $\tilde{B}$
axis for tetrahedral $\tilde{B}$
267 facets
575 facets
1396 facets

Figure 3.33: Mesh instances of the Venus model (left), seam and jump sheet structures (middle) and medial axes as non-manifold, piece-wise linear meshes (right).



triangular mesh | axis for tetrahedral $\tilde{B}$ | offset for tetrahedral $\tilde{B}$

860 facets

1814 facets

3490 facets

Figure 3.34: Various mesh instances of the Chinese-lion model with medial axes and offsets.

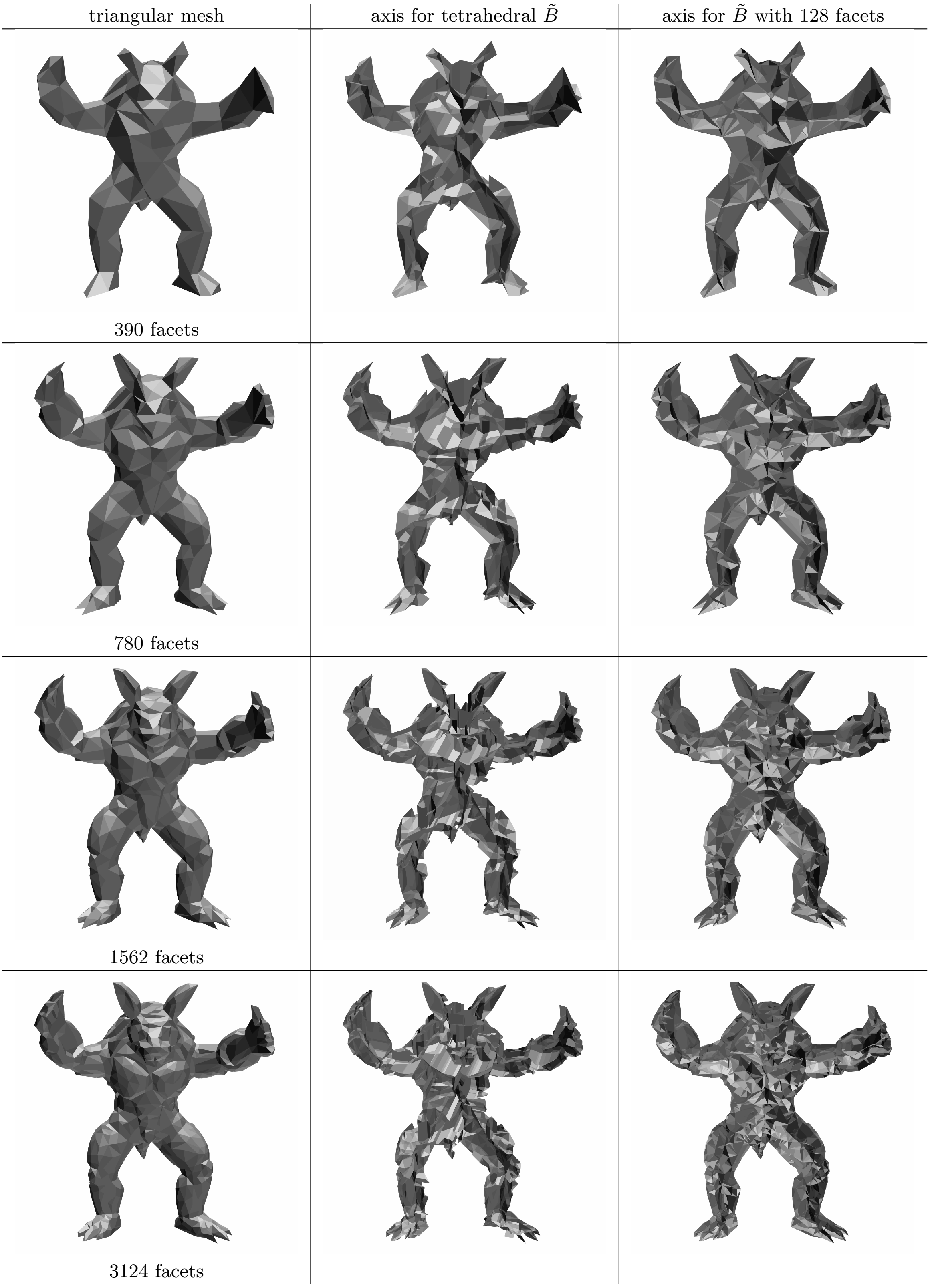

triangular mesh
axis for tetrahedral $\tilde{B}$
axis for $\tilde{B}$ with 128 facets
390 facets
780 facets
1562 facets
3124 facets

Figure 3.35: Various instances of the Armadillo mesh and their medial axes.

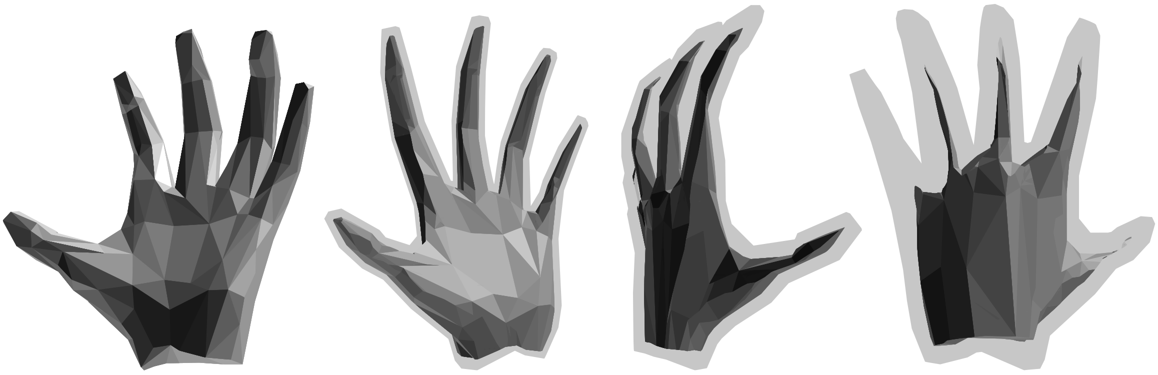


Figure 3.36: These trimmed offset examples, as well as the offsets and medial axes of all other example meshes in the previous chapter have been computed with the algorithm and the implementation described in Section 3.3.2 and Section 3.3.4.

3.4 Conclusion of Chapter 3

Exact medial axis computation is, especially in space, a challenging task. In the Euclidean case, already for piece-wise linear boundary representations (as triangular meshes), bisectors of higher algebraic complexity occur, resulting in even more complex trisectors and overall in a problem being intricate concerning numerical as well as structural issues.

The use of piece-wise linear quasi-metrics has, to our knowledge, up to this point not been directly considered for medial axis computation.[7] As shown in this chapter, the benefits of exact computation with respect to such a metric are numerous.

For triangulated solids, the medial axis with respect to a piece-wise linear quasi-metric is again piece-wise linear. This is one of the reasons why the algorithm presented herein recommends itself for implementation, as for a boundary that is given with rational coordinates, all computations can be done within the set of rational numbers. This guarantees a numerically stable and robust implementation, and allows an exact representation of the medial axis.

To handle the combinatorial and structural complexity of the axis, its three-dimensional components are projected to two-dimensional contact arrangements which represent the complete structure. The algorithm works solely on these contact arrangements, thus reducing a three-dimensional problem to a number of intuitively solvable two-dimensional ones. The completeness of the representation is guaranteed due to the fact that every maximal* ball of $\mathrm{MAT}_{\tilde{B}}(\Omega)$ being centered on the medial axis must have at least one regular contact (or at least projects on the boundary of a respective contact domain). By computing the contact arrangements of all occuring regular typed contact classes, all maximal* balls of $\mathrm{MAT}_{\tilde{B}}(\Omega)$ are covered. The implementation is also easily adaptable for parallel data processing, as the above mentioned contact arrangements can be computed independently from one another.

By assembling the medial axis components, a representation as a non-manifold triangular mesh is obtained, which then allows convenient trimmed offset computation with respect to the underlying quasi-metric.

It might be no big surprise, that the medial axis with respect to a piece-wise linear quasi-

---

[7] Approaches for Voronoi computation with respect to different convex distance functions exist but are not very elaborate or hard to implement [25, 26, 67].

metric induced by a unit ball $\tilde{B}$ converges towards the Euclidean axis if $\tilde{B}$ converges towards the Euclidean unit ball.

But the manipulation of the unit ball raises several other interesting questions, being the main issues for ongoing work in this context. Given a mesh, what does a (preferably combinatorially small) polyhedral unit ball have to look like to reduce the occurrence of pseudo-seams? With a decreasing number of pseudo-seams, a combinatorial structure close to the Euclidean medial axis is to be expected. On the other hand the implicit pruning induced by the piecewise linear metric might be a welcome feature. This leads to the question how to locate points on the unit sphere, such that the vertices of the resulting convex polyhedral ball enter as many flat convex features of a mesh as possible. Modifications of the unit ball $\tilde{B}$ do affect the geometric as well as the combinatorial appearance of $\mathrm{MA}_{\tilde{B}}(\Omega)$. Another interesting task for future research is to identify and isolate the combinatorially stable – and thus essential – parts of the medial axis by comparing the representations for different quasi-metrics $\delta_{\tilde{B}}$ resulting from several different polyhedral unit balls $\tilde{B}$.

Chapter 4

Summary and future work

The main motivation for the work presented in this thesis, was to provide more than theoretical concepts and bounds of abstract algorithms for medial axis computation. The emphasis was more on the practicability and the implementation of the algorithms, as well as their flexibility and applicability for related problems. In particular we wanted to point out and exploit the strong link between the boundary representation of a shape and its medial axis transform. Together with the requirement of numerical stability these considerations naturally suggested the use of a powerful library as CGAL [90], which already provides many basic geometric tools, allowing the user to focus on the more sophisticated problems.

The initial problem treated in Chapter 2 was to compute the exact medial axis of shapes with smooth circular boundary representations. This led to the development of four base cases, to which every shape can be reduced by decomposing it via the footpoints of maximal disks, already exploiting the connection between the medial axis transform and the boundaries shape.

By adapting parts of the algorithm, the original four base cases were supplemented by nine more, extending the set of processable shapes to general circular boundaries.

The next logical step was to extend the set of valid shapes to non-simple ones, enclosing holes. This was achieved by augmenting such a shape at specific locations, thus enabling us to handle it like a combinatorially simple one. When interpreting the occuring holes as Voronoi sites, the resulting structure is equivalent to the edge graph of an extended Voronoi diagram. Self-edges can be pruned, by having a look at the boundary primitives associated with them. Again we take advantage of the link between the axis and associated boundary structures.

However, where this relation of the axis and the boundary really shows its potential is trimmed offset computation. Especially convenient is the fact that the partial offsets of the base cases can be merged by simple concatenation. In general, the concept of divide-and-conquer proved to be very efficient and flexible for all the above problems, as virtually all implementation issues could be handled on the low complexity level of the occuring base cases.

Efforts to extend the divide-and-conquer approach from the 2D to the 3D case mainly failed due to two major drawbacks. Not only is the definition and identification of base cases a difficult task, but also the curves on the boundary at which we would have to split are of higher algebraic complexity. Consequently, the philosophy behind the approach chosen in Chapter 3 was different than the one for the 2D one. While the algorithm in Chapter 2 computes the exact medial axis of a circular boundary representation with respect to the

Euclidean metric, we adapted the underlying metric in the 3D case to obtain a piece-wise linear medial axis for triangulated solids.

In the approach from Chapter 2, by letting the circular boundary representation converge towards the original shape, the medial axis of arc approximation converges towards the medial axis of the original shape. The algorithm in Chapter 3 computes the exact medial axis of a triangulated solid, but with respect to a piece-wise linear distance function. By letting the metric-inducing polyhedral unit ball converge towards the Euclidean unit ball, the resulting piece-wise linear medial axis converges towards the Euclidean one.

Medial axis computation for the triangulated solid is now not achieved by means of divide-and-conquer and base cases, but by building projection arrangements on the planes resulting from contacts between the solids boundary and the polyhedral unit ball. These are now, due to the piece-wise linear metric, composed of simple line segments. Again, the medial axis is constructed with help of its relation to the boundary of the respective shape.

This again results in a set-up which is very convenient for the computation of trimmed inner offsets, of course with respect to the piece-wise linear distance function. Unlike in the 2D approach, however, not base-case-shapes are the smallest subproblems, but single axis sheets represented by arrangement faces are to be handled directly.

Although the above algorithms have alread gone through a considerable development in the course of their extension history, the possibilities for ongoing work are far from being exhausted.

The 2D-algorithm from Chapter 2 certainly recommends itself (concerning its basic divide-and-conquer mechanism) for application on shapes with algebraically more complex boundaries. This would cause an increase in the number of base cases that have to be considered, and bisectors with higher algebraic degree, what will be examined in future work.

The approach from Chapter 3, dealing with piece-wise linear metrics and triangulated solids seems even more interesting for further research. Convergence of the induced piece-wise linear medial axis to the Euclidean one could so far only be proven for the planar case (see [3]). Still missing as well is an upper bound for the combinatorial size of the axis in 3D, with respect to the size of the shapes and the unit balls boundary. Beside these rather theoretical issues, also the pruning effect caused by a polygonal or polyhedral unit ball deserves more attention in ongoing work, for more details see Section 3.4. The main question is: What does a unit ball have to look like, which maximizes or minimizes this implicit pruning for a given shape? A closer look at the distance relation between a polyhedron-induced and the Euclidean medial axis is also a promising topic and may yield interesting results.

All in all we hope for this thesis to have shown, that the medial axis, despite it already being the topic of numerous scientific works, still deserves attention in future research due to its thematic diversity and varied applicability.

Glossary

AABB tree data structure that speeds up detection of intersections in 3D, 73

`arrang` algorithm for computation of the contact arrangement of a contact class, 83

augmented domain domain in the plane enhanced by an augmenting disk, 31

augmenting disk maximal disk whose center point breaks cycle of the medial axis, 31

auxiliary arc circular arc derived from a dividing disk, 12

base case partial shape composed of circular arcs, whose medial axis can be computed directly, 13

bisector a bisector between two sites is the locus of points that are equidistant to both sites, 42

branching point point on the medial axis in 2D with three footpoints, 8

CGAL C++ library with geometric and numerical tools, 21

circular arc boundary boundary which is composed of circular arcs, 10

contact pair consisting of one component of a shapes boundary, and one component of a maximal* balls boundary, 54

contact arrangement arrangement of projection lines, 70

contact class set of contacts sharing the shape component, and whose ball component is derived from the same component of the underlying unit ball, 54

contact domain set of projections on the contact plane, associated with a contact class, 68

contact plane plane spanned by the components of a regular contact class, 67

contact type describes the type of components forming a contact, 53

convex in a convex feature of a shape all line segments connecting two points that lie in this feature are contained in the shape, 14

cycle in this context: closed curve in a line graph, 30

dividing disk maximal disk which induces split of the medial axis into combinatorially smaller partial axis, 12

footpoint point shared by a maximal inscribed or maximal* ball and the boundary of a shape, 7

Gmpq exact rational number type, 21

Halfedge data type used for representing a mesh in the halfedge structure, 92

Hausdorff distance distance measure between two arbitrary sets in the same space, 7

hole set of points bounded by a boundary part of a shape, which is not part of this shape; here also considered as a Voronoi site, 29

jump edge special medial axis edge in 2D that results from piece-wise linear quasi-metrics, 55

jump sheet special medial axis sheet in 3D that results from piece-wise linear quasi-metrics, 57

junction point point on the medial axis in 3D with four footpoints, 8

leaf point point on the medial axis in 2D associated with local curvature maximum, 8

local maximal curvature point / arc on the boundary of shape in 2D, which describes a local maximum of curvature, 18

`maxdisk` algorithm for computation of a maximal disk of a circular arc boundary shape, 13

maximal* ball ball contained in a shape that shares at least two points with its boundary, 52

maximal disk maximal inscribed ball in the plane, 9

maximal inscribed ball ball which is maximal with respect to inclusion in a shape, 7

`medax` algorithm for computation of the medial axis of a circular arc boundary shape, 20

medial axis shape descriptor composed of the center points of maximal inscribed balls, 7

medial axis transform set of all maximal inscribed balls of a shape, 7

mesh polygonal net, mostly composed of triangles, 51

offset curve whose points are at a fixed normal distance of a given curve, 7

opposite contact maximal* ball centered on sheet projects in two contact domains, one contact is called the opposite contact of the other one and vice versa, 72

original arc circular arc derived from an arc on the boundary of a shape, 12

partial axis medial axis of a part of a shape, obtained by decomposition, 18

piece-wise linear structure composed of straight lines and planes, 51

projection projection of a maximal* ball in a contact domain, 68

projection line projection of a seam, pseudo-seam or jump sheet in a contact domain, 76

`projline` algorithm for computation of a projection line in a contact domain that crosses a given segment, 80

pruning thinning effect of the medial axis resulting from pseudo-structures, 60

pseudo-branching branching on the medial axis in 2D which has only two footpoints, 55

pseudo-seam a seam in 3D represented by set of maximal* balls which have only two footpoints, 57

quasi-metric metric without symmetry property, 51

rasqex rational number type extended by the square-root operator, 40

RCAB circular arc boundary composed of rational arcs, 41

reflex concave feature, opposite of convex, 14

seam set of points on the medial axis in 3D with three footpoints, 8

`searchcc` binary search algorithm for new opposite contact, 77

self-edge edge of Voronoi diagram representing maximal disks which have two footpoints on the same site, 29

shape connected set in two or three dimensions, 7

sheet set of points on the medial axis in 3D with two footpoints, 8

`sweep` algorithm for intersection detection along an edge of a contact arrangement, 78

tree connected, cycle-free line graph in the plane, 8

triangulated solid simple polygonal area in 2D, connected shape bounded by a triangular mesh in 3D, 51

trimmed offset offset whithout self-intersecting parts, 7

unit ball convex set, containing the origin, that induces a distance function, 7

Voronoi diagram decomposition of a space determined by distances to a discrete set of sites, 29

Bibliography

[1] O. Aichholzer, W. Aigner, F. Aurenhammer, T. Hackl, B. Jüttler, E. Pilgerstorfer, and M. Rabl. Divide-and-conquer for Voronoi diagrams revisited. *Computational Geometry: Theory and Applications*, 43:688–699, 2010.

[2] O. Aichholzer, W. Aigner, F. Aurenhammer, T. Hackl, B. Jüttler, and M. Rabl. Medial axis computation for planar free-form shapes. *Computer-Aided Design*, 41:339–349, 2009.

[3] O. Aichholzer, W. Aigner, F. Aurenhammer, and B. Jüttler. Exact medial axis computation of triangulated solids with respect to piecewise linear metrics. *Lecture Notes in Computer Science*, to appear 2011.

[4] O. Aichholzer, W. Aigner, T. Hackl, and N. Wolpert. Exact medial axis computation for circular arc boundaries. *Lecture Notes in Computer Science*, to appear 2011.

[5] O. Aichholzer, F. Aurenhammer, T. Hackl, B. Jüttler, M. Oberneder, and Z. Sír. Computational and structural advantages of circular boundary representation. In *Lecture Notes in Computer Science*, volume 4619, pages 374–385, 2007.

[6] O. Aichholzer, F. Aurenhammer, B. Kornberger, S. Planting, G. Rote, A. Sturm, and G. Vegter. Recovering structure from r-sampled objects. In *Proceedings of the Symposium on Geometry Processing '09*, pages 1349–1360, 2009.

[7] W. Aigner. The medial axis of planar shapes. Master's thesis, University of Technology, Graz, 2007.

[8] V. R. Algazi and J. W. Brandt. Continuous skeleton computation by Voronoi diagram. *CVGIP: Image Understanding*, 55:329–338, 1991.

[9] P. Alliez, S. Tayeb, and C. Wormser. AABB tree. In *CGAL User and Reference Manual.* CGAL Editorial Board, 3.7 edition, 2010.

[10] H. Alt, O. Cheong, and A. Vigneron. The Voronoi diagram of curved objects. *Discrete & Computational Geometry*, 34:439–453, 2005.

[11] H. Alt and C.-K. Yap. Motion planning in the CL-environment. In *Lecture Notes In Computer Science*, volume 382, pages 373–380. 1989.

[12] N. M. Amato, P. F. Stiller, and S. A. Wilmarth. Motion planning for a rigid body using random networks on the medial axis of the free space. In *Proceedings of the 15th annual symposium on Computational geometry '99*, pages 173–180, New York, USA, 1999. ACM.

[13] T. Asano, J. Matousek, and T. Tokuyama. The distance trisector curve. *Advances in Mathematics*, 212(1):338–360, 2007.

[14] D. Attali, J.-D. Boissonnat, and H. Edelsbrunner. Stability and computation of medial axes: a state of the art report. In B. Hamann T. Moeller and B. Russell, editors, *Mathematical Foundations of Scientific Visualization, Computer Graphics and Massive Data Exploration.* Springer-Verlag, Mathematics and Visualization, 2007.

[15] B. Bastl, B. Jüttler, J. Kosinka, and M. Lávička. Volumes with piecewise quadratic medial surface transforms: Computation of boundaries and trimmed offsets. *Computer-Aided Design*, 42:671–679, 2010.

[16] E. Berberich, A. Eigenwillig, M. Hemmer, S. Hert, K. Mehlhorn, and E. Schömer. A computational basis for conic arcs and boolean operations on conic polygons. In *ESA 2002, Lecture Notes in Computer Science*, pages 174–186, 2002.

[17] H. Blum. A transformation for extracting new descriptors of shape. In Weiant Wathen-Dunn, editor, *Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press, Cambridge, England, 1967.

[18] J.-D. Boissonnat and S. Oudot. Provably good sampling and meshing of surfaces. *Graphical Models*, 67:405–451, 2005.

[19] E. Brisson, N. M. Patrikalakis, and E. C. Sherbrooke. An algorithm for medial axis transform of 3-D polyhedral solids. *IEEE Trans. on Visualization and Computer Graphics*, 2:44–61, 1996.

[20] D. Brunner and G. Brunnett. Automatic bone generation for character animation using the discrete medial axis transformation. *Tagungsband Virtuelle und Erweiterte Realität, 1. Workshop der GI-Fachgruppe VR/AR*, pages 339–350, 2004.

[21] C. Burnikel. Rational points on circles. Technical report, Max-Planck-Institut für Informatik, 1998.

[22] C. Burnikel, S. Funke, K. Mehlhorn, S. Schirra, and S. Schmitt. A separation bound for real algebraic expressions. In *ESA 2001, Lecture Notes in Computer Science*, pages 154–265, 2001.

[23] L. Cao and J. Liu. Computation of medial axis and offset curves of curved boundaries in planar domain. *Computer-Aided Design*, 40:465–475, 2008.

[24] F. Chazal and A. Lieutier. The λ-medial axis. *Graphical Models*, 67:304–331, 2005.

[25] L. P. Chew and R. L. Drysdale. Voronoi diagrams based on convex distance functions. In *Proc. 1st Ann. ACM Symposium on Computational Geometry*, pages 235–244, 1985.

[26] L. P. Chew, K. Kedem, M. Sharir, B. Tagansky, and E. Welzl. Voronoi diagrams of lines in 3-space under polyhedral convex distance functions. *Journal of Algorithms*, 29:238–255, 1998.

[27] F. Chin, J. Snoeyink, and C. A. Wang. Finding the medial axis of a simple polygon in linear time. In *Discrete Computational Geometry*, pages 382–391. Springer-Verlag, 1995.

[28] H. I. Choi, S. W. Choi, and H. P. Moon. Mathematical theory of medial axis transform. *Pacific Journal of Mathematics*, 181(1):57–88, 1997.

[29] H. I. Choi and C. Y. Han. The medial axis transform. In J. Hoschek G. Farin and M. S. Kim, editors, *The Handbook of Computer Aided Geometric Design*, pages 451–471. Amsterdam, The Netherlands, 2002.

[30] J. J. Chou. Voronoi diagrams for planar shapes. *IEEE Computer Graphics and Applications*, 15:52–59, 1995.

[31] J.-M. Chung and N. Ohnishi. Matching and recognition of planar shapes using medial axis properties, 2007.

[32] E. Cohen, S. Drake, and G. Elber. Medial axis transform toward high speed machining of pockets. *Computer-Aided Design*, 37(2):241–250, 2005.

[33] T. Culver, J. Keyser, and D. Manocha. Exact computation of the medial axis of a polyhedron. *Computer-Aided Geometric Design*, 21(1):65–98, 2004.

[34] W. L. F. Degen. Exploiting curvatures to compute the medial axis for domains with smooth boundary. *Computer-Aided Geometric Design*, 21(7):641–660, 2004.

[35] T. K. Dey and W. Zhao. Approximating the medial axis from the Voronoi diagram with a convergence guarantee. *Algorithmica*, 38:179–200, 2003.

[36] S. J. Dickinson, A. Shokoufandeh, K. Siddiqi, and S. W. Zucker. Shock graphs and shape matching. *International Journal for Computer Vision*, 35:13–32, 1999.

[37] R. L. S. Drysdale and D. T. Lee. Generalization of Voronoi diagrams in the plane. *SIAM Journal on Computing*, 10:73–87, 1981.

[38] H. Du and H. Qin. Medial axis extraction and shape manipulation of solid objects using parabolic pdes. In *Proceedings of the 9th ACM symposium on Solid modeling and applications '04*, pages 25–35, Aire-la-Ville, Switzerland, 2004. Eurographics Association.

[39] Z. Du, C. Li, S. Pion, V. Sharma, and C. Yap. The Core library. http://cs.nyu.edu/exact/core_pages/downloads.html.

[40] T. Dubé and C. K. Yap. The exact computation paradigm. In D.-Z. Du and F. K. Hwang, editors, *Computing in Euclidean Geometry*, volume 1 of *Lecture Notes Series on Computing*, pages 452–492. World Scientific, 1995.

[41] G. Elber, I. Hanniel, M.-S. Kim, and R. Muthuganapathy. Precise Voronoi cell extraction of free-form rational planar closed curves. In *SPM '05: Proceedings of the 2005 ACM symposium on Solid and physical modeling*, pages 51–59, New York, USA, 2005. ACM.

[42] G. Elber and M.-S. Kim. Bisector curves of planar rational curves. *Computer-Aided Design*, 30(14):1089–1096, 1998.

[43] G. Elber, M.-S. Kim, and J.-K. Seong. Trimming local and global self-intersections in offset curves/surfaces using distance maps. *Computer-Aided Design*, 38:183–193, 2006.

[44] I. Z. Emiris and M. I. Karavelas. The predicates of the Apollonius diagram: Algorithmic analysis and implementation. *Computational Geometry*, 33(1-2):18–57, 2006.

[45] I. Z. Emiris, E. P. Tsigaridas, and G. M. Tzoumas. Exact delaunay graph of smooth convex pseudo-circles: general predicates, and implementation for ellipses. In *SPM '09: 2009 SIAM/ACM Joint Conference on Geometric and Physical Modeling*, pages 211–222, New York, USA, 2009. ACM.

[46] I. Z. Emiris and G. M. Tzoumas. Exact and efficient evaluation of the incircle predicate for parametric ellipses and smooth convex objects. *Computer-Aided Design*, 40(6):691–700, 2008.

[47] M. Etzion and A. Rappoport. Computing the Voronoi diagram of a 3-D polyhedron by separate computation of its symbolic and geometric parts. In *Proceedings Symposium on Solid Modeling and Applications*, pages 167–178. ACM, 1999.

[48] H. Everett, C. Gillot, D. Lazard, S. Lazard, and M. Pouget. The voronoi diagram of three arbitrary lines in $\mathbb{R}^3$. In *In Proc. 25th European Workshop on Computational Geometry '09*, pages 297–300, 2009.

[49] R. T. Farouki and R. Ramamurthy. Degenerate point/curve and curve/curve bisectors arising in medial axis computations for planar domains with curved boundaries. *Computer-Aided Geometric Design*, 15(6):615–635, 1998.

[50] R. T. Farouki and R. Ramamurthy. Voronoi diagram and medial axis algorithm for planar domains with curved boundaries i. theoretical foundations. *Journal of Computational and Applied Mathematics*, 102(1):119–141, 1999.

[51] R. T. Farouki and R. Ramamurthy. Voronoi diagram and medial axis algorithm for planar domains with curved boundaries ii: detailed algorithm description. *Journal of Computational and Applied Mathematics*, 102(2):253–277, 1999.

[52] R. Feichtinger, B. Jüttler, and Z. Šír. Approximating curves and their offsets using biarcs and Pythagorean hodograph quintics. *Computer-Aided Design*, 38:608–618, 2006.

[53] E. Fogel, D. Halperin, R. Wein, and B. Zukerman. 2D arrangements. In *CGAL User and Reference Manual*. CGAL Editorial Board, 3.8 edition, 2011.

[54] J. Giesen, B. Miklos, and M. Pauly. Discrete scale axis representations for 3d geometry. *ACM Transactions on Graphics*, 29:101:1–101:10, 2010.

[55] D. Gisch and J. M. Ribando. Apollonius' problem: A study of solutions and their connections, 2004.

[56] H. Nebi Gürsoy and N. M. Patrikalakis. An automatic coarse and fine surface mesh generation scheme based on medial axis transform: Part ii implementation. *Engineering with Computers*, 8:179–196.

[57] H. Nebi Gürsoy and N. M. Patrikalakis. An automatic coarse and fine surface mesh generation scheme based on medial axis transform: Part i algorithms. *Engineering with Computers*, 8:121–137, 1992.

[58] B. Gurumoorthy and M. Ramanathan. Constructing medial axis transform of planar domains with curved boundaries. *Computer-Aided Design*, 35(7):619–632, 2003.

[59] B. Gurumoorthy and M. Ramanathan. Interior medial axis transform computation of 3D objects bound by free-form surfaces. *Computer-Aided Design*, 42:1217–1231, 2010.

[60] M. Held. Voronoi diagrams and offset curves of curvilinear polygons. *Computer-Aided Design*, 30(4):287–300, 1998.

[61] M. Held. Vroni: An engineering approach to the reliable and efficient computation of Voronoi diagrams of points and line segments. *Computational Geometry: Theory and Applications*, 18(2):95–123, 2001.

[62] M. Held and S. Huber. Topology-oriented incremental computation of Voronoi diagrams of circular arcs and straight-line segments. *Computer-Aided Design*, 41:327–338, 2009.

[63] M. Held, G. Lukacs, and L. Andor. Pocket machining based on contour-parallel tool paths generated by means of proximity maps. *Computer-Aided Design*, 26:189–203, 1994.

[64] M. Hemmer. Algebraic foundations. In *CGAL User and Reference Manual*. CGAL Editorial Board, 3.7 edition, 2010.

[65] D. Hoey and M. I. Shamos. Closest-point problems. In *Proc. 16th Ann. Symp. Foundations of Computer Science*, pages 151–162, 1975.

[66] I.-K. Hwang, D.-S. Kim, and B.-J. Park. Representing the voronoi diagram of a simple polygon using rational quadratic bézier curves. *Computer-Aided Design*, 27:605–614, 1995.

[67] C. Icking, R. Klein, N.-M. Lê, L. Ma, and F. Santos. On bisectors for convex distance functions in 3-space. In *In Proc. 11th Canadian Conference on Computational Geometry*, 1999.

[68] T. Jones. Geomview. *Linux Journal*, 1996. http://www.geomview.org/.

[69] M. Karavelas. 2D segment Delaunay graphs. In *CGAL User and Reference Manual*. CGAL Editorial Board, 3.7 edition, 2010.

[70] J. L. Kelley and I. Namioka. *Linear topological spaces*. Springer, New York, USA, 1976.

[71] L. Kettner. 3D polyhedral surfaces. In *CGAL User and Reference Manual*. CGAL Editorial Board, 3.8 edition, 2011.

[72] L. Kettner. Halfedge data structures. In *CGAL User and Reference Manual*. CGAL Editorial Board, 3.8 edition, 2011.

[73] D. G. Kirkpatrick. Efficient computation of continuous skeletons. In *Proc. 20th Ann. Symp. Foundations of Computer Science*, pages 18–27, 1979.

[74] R. Klein, K. Mehlhorn, and S. Meiser. Randomized incremental construction of abstract Voronoi diagrams. *Computational Geometry: Theory and Applications*, 3:157–184, 1993.

[75] P. Kunkel. The tangency problem of Apollonius: three looks. *BSHM Bulletin: Journal of the British Society for the History of Mathematics*, 22:34–46, 2007.

[76] D. T. Lee. Medial axis transformation of a planar shape. *IEEE Pattern Analysis and Machine Intelligence*, vol. PAMI-4:363–369, 1982.

[77] Y.-G. Lee and K. Lee. Computing the medial surface of a 3-D boundary representation model. *Advances in Engineering Software*, 28:593–605, 1997.

[78] L. Linsen. Point cloud representation. Technical report, Fakultät f. Informatik, Universität Karlsruhe, 2001.

[79] D. S. Meek and D. J. Walton. Approximating smooth planar curves by arc splines. *Journal of Computational and Applied Mathematics*, 59(2):221–231, 1995.

[80] D. S. Meek and D. J. Walton. Spiral arc spline approximation to a planar spiral. *Journal of Computational and Applied Mathematics*, 107(1):21–30, 1999.

[81] K. Mehlhorn and S. Näher. *The LEDA platform for combinatorial and geometric computing.* Cambridge University Press, 1999.

[82] A. Schinzel and W. Sierpinski. Elementary theory of numbers (second edition). *North-Holland mathematical library*, 31, 1988.

[83] M. Sharir. Intersection and closest-pair problems for a set of circular discs. *SIAM Journal on Computing*, 14:448–468, 1985.

[84] K. Weiler. The radial-edge structure: A topological representation for non-manifold geometric boundary representations. *Geometric Modelling for CAD Applications*, pages 3–36, 1988.

[85] W. A. Wilson. On quasi-metric spaces. *American Journal of Mathematics*, 53(3):675–684, 1931.

[86] C.-K. Yap. An $O(n \log n)$ algorithm for the Voronoi diagram of a set of simple curve segments. *Discrete & Computational Geometry*, 2:365–393, 1987.

[87] z. Boost C++ libraries. http://www.boost.org/.

[88] z. Mozilla. http://www.mozilla.org/.

[89] z. Qt, cross-platform application framework. http://trolltech.com/products/qt/.

[90] z. CGAL, Computational Geometry Algorithms Library. http://www.cgal.org/.

[91] z. GMP, GNU Multiple Precision Arithmetic Library. http://gmplib.org/.

Curriculum Vitae of Wolfgang Aigner

August 2, 2011

Affiliation

Dipl.-Ing. Wolfgang Aigner
Institute for Software Technology
Graz University of Technology
Inffeldgasse 16b, A-8010 Graz, Austria
Email: `waigner@ist.tugraz.at`

Personal Data

Born 10th of August, 1982 in Tamsweg (Austria)
Austrian nationality

Education

Graz University of Technology, Austria
2007 - present
PhD student, Adviser: Prof. O. Aichholzer

Graz University of Technology, Austria
2001 - 2007
Dipl.-Ing. (MSc), Technical Mathematics, Adviser: Prof. F. Aurenhammer
Studies of Technical Mathematics (with emphasis on informatics), Diploma Thesis at the Institute for Theoretical Computer Science (with distinction)

High School, Tamsweg, Austria
1992 - 2000
Matura (A-levels) in mathematics, english, german, latin and philosophy and psychology (with distinction)

Career History

Graz University of Technology, Austria
2011 - present
Institute for Theoretical Computer Science
Research Assistant, supported by the Austrian FWF 'EuroGIGA Voronoi'

Graz University of Technology, Austria
2007 - 2009 and 2010 - 2011
Institute for Software Technology
Research Assistant, supported by the Austrian FWF National Research Network 'Industrial Geometry' S9205-N12

Johannes Kepler University Linz, Austria
2009 - 2010
Institute of Applied Geometry
Research Assistant, supported by the Austrian FWF National Research Network 'Industrial Geometry' S9205-N12

Grazer Wechselseitige Versicherung AG, Austria
2005 - 2007
Part-time employee as insurance mathematician in the headquarters in Graz, Austria

Other Occupations

Comenius Education Project
2006 - 2007
Free programmer (Mathematica)

Austrian Armed Forces
2000 - 2001
National service (8 months)

Research Interests

Data structures, algorithms, computational geometry and graph theory in general, medial axis and offset computation, numerical exact computation in particular, software development in C++ (CGAL) and various other programming environments.

Publications Overview

2 articles in refereed journals.
3 [+1 accepted, +2 to appear] articles in refereed proceedings.
1 poster presentation.
Diploma Thesis.

Publications

Articles in refereed journals (thesis-relevant marked bold):

O. Aichholzer, W. Aigner, F. Aurenhammer, T. Hackl, B. Jüttler, E. Pilgerstorfer, and M. Rabl. Divide-and-conquer for Voronoi diagrams revisited. *Computational Geometry: Theory and Applications*, 43(8):688–699, 2010. Special Issue on the 25th Annual Symposium on Computational Geometry (SoCG'09).

O. Aichholzer, W. Aigner, F. Aurenhammer, T. Hackl, B. Jüttler, and M. Rabl. Medial axis computation for planar free-form shapes. *Computer-Aided Design*, 41(5):339–349, 2009. Special issue: Voronoi Diagrams and their Applications.

Articles in refereed proceedings (thesis-relevant marked bold):

O. Aichholzer, W. Aigner, F. Aurenhammer, K.Č Dobiášová, B. Jüttler and G. Rote. Triangulations with circular arcs. 19^{th} *Symp. on Graph Drawing 2011*, accepted.

O. Aichholzer, W. Aigner, F. Aurenhammer, and B. Jüttler. Exact medial axis computation of triangulated solids with respect to piecewise linear metrics. *Lecture Notes in Computer Science*, to appear 2011.

O. Aichholzer, W. Aigner, T. Hackl, and N. Wolpert. Exact medial axis computation for circular arc boundaries. *Lecture Notes in Computer Science*, to appear 2011.

O. Aichholzer, W. Aigner, F. Aurenhammer, K.Č Dobiášová, and B. Jüttler. Arc triangulations. In *Proc.* 26^{th} *European Workshop on Computational Geometry EuroCG '10*, pages 17–20, Dortmund, Germany, 2010.

O. Aichholzer, W. Aigner, F. Aurenhammer, T. Hackl, B. Jüttler, E. Pilgerstorfer, and M. Rabl. Divide-and-conquer for Voronoi diagrams revisited. In 25^{th} *Ann. ACM Symp. Computational Geometry*, pages 189–197, Aarhus, Denmark, 2009.

O. Aichholzer, W. Aigner, F. Aurenhammer, T. Hackl, B. Jüttler, E. Pilgerstorfer, and M. Rabl. Divide-and-conquer for Voronoi diagrams revisited. In *Proc.* 25^{th} *European Workshop on Computational Geometry EuroCG '09*, pages 293–296, Brussels, Belgium, 2009.

Poster presentations:

W. Aigner. Exact medial axis computation for circular arc boundaries. Poster presentation at Curves & Surfaces 2010, Avignon, 2010.

Diploma thesis:

W. Aigner. The medial axis of planar shapes. Master's thesis, IGI-TU Graz, Austria, 2007.