Geometric Approximations in 2- and 3-dimensional Space

Dissertation zur Erlangung des Doktorgrades

vorgelegt am

Fachbereich Mathematik und Informatik
der Freien Universität Berlin

2009

von

Astrid-Utte Sturm

Institut für Informatik

Freie Universität Berlin

Takustraße 9

14195 Berlin

sturm@inf.fu-berlin.de

gefördert durch die EU im Rahmen des Forschungsprojektes

ACS, unter Vertragsnummer IST-006413

Betreuer: Prof. Dr. Günter Rote
Institut für Informatik
Freie Universität Berlin
Takustraße 9
D–14195 Berlin
Germany
rote@inf.fu-berlin.de

Gutachter: Prof. Dr. Günter Rote
Institut für Informatik
Freie Universität Berlin
Takustraße 9
14195 Berlin
Germany
rote@inf.fu-berlin.de

Prof. Dipl.-Ing. Dr. Martin Held
Fachbereich Computerwissenschaften
Universität Salzburg
Jakob-Haringer Str. 2
A-5020 Salzburg
Austria
held@cosy.sbg.ac.at

Vorlage zur Begutachtung: 30. Juni 2009
Termin der Disputation: 27. November 2009
Fassung vom: 14. Dezember 2009

Contents

Preface

In this thesis we study approximation problems for curves and surfaces. The goal of geometric approximation is to replace a given complex geometric object by a simpler object while capturing the significant features of the original. If the original object is given by a set of sample points, this problem is also called reconstruction. Approximating and reconstructing objects is a problem that arises in many applications such as computer graphics, computer vision, medical imaging and cartography, to name only a few. The problem goes back decades and is one of the major challenges in computational geometry.

For a *polygonal* curve an approximation can be done either by a simpler polygonal curve (a curve with less segments) or by a higher order curve. The approximation of polygonal curves is a wide topic and many results have already been presented in the past. There is a wide range of publications on the approximation of polygonal curves or ordered set of points by polygonal curves, for example, the publications of Imai and Iri [51, 52], or Guibas and Hershberger [44], to mention only a few. There are two types of optimization problems associated with the polygon approximation problem:

- Min-# problem: Given $\varepsilon \geq 0$, construct an approximate curve with "error" within ε and having the minimum number of line segments.

- Min-ε problem: Given m, construct an approximate curve consisting of at most m line segments with minimum approximation "error".

As these optimization problems were answered for polygonal approximation, the same questions arise for approximation with curves of higher order. The approximation of polygonal curves with curves of higher order, especially with arcs and biarcs, has been mostly heuristic [38, 47, 58, 59, 60, 66, 68, 77].

Higher-order approximations of polygonal curves come into place in areas as, e.g., computer-aided manufacturing environments, geometric modeling and robot path planing. One major task is to smooth the path of, e.g., cutting machines or robots. For example in computer-aided manufacturing environments, tool paths are usually made of line segments and circular arcs, see Mekk and Walton [58, 59, 60]. Now the question of how to approximate polygonal curves with circular arcs or biarcs with certain guarantees arises. We were able to answer the min-# problem for approximating open polygonal curves with circular arcs

and also for biarcs.

The approximation of surfaces and, especially, the reconstruction from scattered data points has received a lot of attention in the past. The problem of reconstructing a surface from a set of sample points is ill-posed by nature. Without certain constraints the given point cloud can be interpolated by infinitely many shapes with differing topology. The notion of a sufficiently dense sample, the r-sample, introduced by Amenta and Bern [7], introduces a constraint on the input data set. With this constraint the corresponding set of reconstructible shapes have all the same topology type as the surface. A large range of algorithms are based on the r-sample theory, see for example [9, 7, 10, 11, 30, 23, 41, 76], again naming only a few. The Delaunay triangulation of the input point set plays another major role in all of these methods, as it does in the α-shape theory of Edelsbrunner [37]. The α-shape is a subcomplex of the Delaunay triangulation of the input point set. If the circumsphere of a Delaunay simplex (vertex, edge, face) has radius at most α, the simplex belongs to the α-shape of the input point set. The optimal value for α depends on the sampling density, therefore the α-shape algorithm works best for uniformly sampled surfaces. Our approach to surface reconstruction, which is similar to the work of Chazal and Lieutier [23], is also based on Delaunay triangulation of the input point set, and we require that the sampling is an r-sample. We construct an approximating polytope P that uses a subset of the input sample points as its vertices and preserves the topology of the sampled surface. Our goal is, on the one hand, to use as few points of the input set as possible and, on the other, to get a flexible approximation with a level of detail that can be tuned from coarse to fine. In contrast to [23], where prior knowledge of the local feature size of the sampled surface is assumed, we obtain an estimation of this function from the data, by using distances to poles [7] (certain vertices of the Voronoi diagram for the input sample). Using a tailored technique of pruning the surface balls, we obtain a coarse-to-fine approximation of the surface by polytopes. This is the first result that uses, from a practical point of view, approximations of local feature size and medial axis to obtain locally adaptive reconstructions of an unknown surface.

Overview

The work is organized in two parts which can be read independently of each other. The introduction to each of the parts gives a short problem description and motivation.

Part I. In the first part of the thesis we deal with approximating polygonal curves in 2-dimensional space.

In Chapter 1 we answer the min-# problem for approximating open polygonal

curves with circular arcs. We prove the following result.

Theorem. *Given an open polygonal curve $P = (p_1, \ldots, p_n)$, a polygonal tolerance boundary of size $O(n)$, and a gate for each p_i, we can approximate P by a minimum number of valid circular arcs in $O(n^2 \log n)$ time and $O(n)$ space.*

A polygonal tolerance boundary is the boundary of a simple polygon that encloses our polygonal curve P. The gates are segments through the points p_i of P, each gate crosses P. The precise formulation for the tolerance boundary and the gates is given in Chapter 1.

In Chapter 2 we answer the min-# problem for approximating open polygonal curves for biarcs and prove the following result:

Theorem. *Given an open polygonal curve $P = (p_1, \ldots, p_n)$, a polygonal tolerance boundary of size $O(n)$, a gate and a tangent direction for each p_i, we can approximate P by a minimum number of valid biarcs in $O(n^2 \log n)$ time and $O(n^2)$ space.*

Part II of this thesis consists of three chapters.

In the second part of the thesis we move on to the 3-dimensional space and to polytope approximations. We initiate the study of this problem by considering convex surfaces only, for simplicity, before moving on to non-convex surfaces. A first natural step to higher-order approximation of convex polytopes is the approximation with spheres or spherical patches.

In Chapter 3 we can show that deciding the existence of an approximation of a convex polytope with a given upper error bound ε and not more than a given number of spherical patches is NP-hard.

In Chapter 4 we present a new technique for constructing a curved surface based on inscribed polytopes resulting in a convex surface consisting of spherical patches.

To tackle the approximation problem for non-convex polytopes we pick up the idea of an incremental approximation algorithm introduced in Chapter 4. This induces the problem of finding a simple and topologically correct start polytope, the seed polytope, for non-convex polytopes.

In Chapter 5 we describe how to construct for a surface in 3D space, given by sample points S, a coarse approximating polytope P. P uses a subset of the points as vertices and preserves the topology. In contrast to surface reconstruction we do not use all sample points, but try to use as few points as possible. We also show how the method can be used to construct triangular surfaces from point clouds in a scalable manner.

Acknowledgments

I would like to thank all the people that have supported me in writing this thesis. First of all, I would like to thank my advisor, Günter Rote, for the great guidance and intensive research, for many good ideas and excellent discussions, for being very patient and understanding, tolerant and supporting. Due to his patience and support, I was able to finish this thesis despite three kids. I thank all colleagues from the work group "Theoretical Computer Science" at Freie Universität Berlin for their generous help and support, for many fruitful discussions, for the friendly atmosphere and excellent research environment. Special thanks to Ludmila Scharf, who shares not only an office with me, but also is a great help discussing research, putting up with VPN unpredictability, Apple online help etc., etc.. I thank Martin Held for the fruitful discussions regarding the biarcs. I am grateful to all participants of the ACS-project for the excellent research environment and for the interesting discussions. Many thanks go to my co-authors for sharing their knowledge and experience with me, and for many fruitful and interesting discussions: Oswin Aichholzer, Franz Aurenhammer, Kevin Buchin, Scot Drysdale, Bernhard Kornberger, Simon Plantinga, Günter Rote and Gert Vegter. I want to specially thank Simon Plantinga and Gert Vegter for the great research stays in Groningen, which were always very inspiring. Thanks also to Bernhard Kornberger for implementing our surfaceballs algorithm. I thank the 'Conors' for their generous help and support in all smaller and bigger 'crisis' and especially Anne for proof-reading this thesis. I am grateful to my family and all my friends for their friendship and their support and especially Anneliese for taking care of my family every time I attended a conference. Finally, and most importantly, I thank my husband Andreas and my kids, Anna, Maximilian and Florian, for their support and love. With you life never gets boring!

Berlin, June 2009.

Part I

Approximation of polygonal curves in 2-dimensional space

Introduction

A polygonal curve in 2-dimensional space, which is given by an ordered set of points, can either be approximated with a polygonal curve with less segments or by a higher-order curve. The approximation algorithms can be further categorized by the optimization problems they solve (min-# problem and min-ε problem, see Preface) and the way they choose vertices for the approximating curve. Further there are several error criteria defined for the approximation with a polygonal curve with less segments, the error criteria refer to the way the error of an approximating segment is measured.

The motivation for approximating a polygonal curve is based on various applications in cartography, computer-aided manufacturing environments, geometric modeling and robot path planing. In engineering design the approximation of data by curves of various forms is an essential step in the design process. One area where higher-order approximations of polygonal curves come into play are computer-aided manufacturing environments. In computer-aided manufacturing environments, tool paths are usually made of line segments and circular arcs, see Meek and Walton [58, 59, 60]. Therefore the question of how to approximate polygonal curves with circular arcs or biarcs arises. Many results achieved in higher-order approximation of polygonal curves are motivated by these computer-aided manufacturing environments, for example, see the results presented by Eibel[38], Held and Eibel [47], Meek and Walton [58, 59, 60], Piegel [66], Schönherr [68] and Yeung and Walton [77].

In contrast to approximation by polygonal curves, the theoretical bounds of these problems are not so well studied, but without theoretical bounds it is difficult to say anything about the quality of these algorithms. There are two types of optimization problems associated with the polygon approximation problem, the min-# and the min-ε problem. As these optimization problems were answered for polygonal approximation, for example by Chan and Chin [20], Guibas and Hershberger [44], Imai and Iri [51, 52], Melkman and O'Rourke [61], Toussaint [73] and Varadarajan [75], the same questions arise for approximation with curves of higher order. We were able to answer the min-# problem for approximating open polygonal curves with circular arcs and also for biarcs. In Chapter 1 we introduce an algorithm for approximating a given open polygonal curve with a minimum number of circular arcs. We present an algorithm which finds a series of circular arcs that approximate the polygonal curve while remaining within a given tolerance region. This series contains the minimum number of arcs of any such series. Our algorithm takes $O(n^2 \log n)$ time for an original polygonal chain with n vertices. Using a similar approach, we design an algorithm with a runtime of $O(n^2 \log n)$, for computing a tangent-continuous approximation with the minimum number of biarcs, for a sequence of points with given tangent directions. This biarcs algorithm is presented in Chapter 2. The results on arc and biarc approximation were published by Drysdale, Rote and Sturm in the journal

Computational Geometry, Theory and Applications in October 2008 [34].

Chapter 1

Approximation with a Minimum Number of Circular Arcs

We assume that we are given an open polygonal curve in two dimensions with a *tolerance region* around the given curve. The *tolerance region* is split into subregions by *gates* through the given points, see Figure 1.1. The precise formulation is given below.

Our algorithm for the optimal approximation by circular arcs determines a subsequence of the input vertices and connects them by a sequence of circular arcs, lying in the tolerance region and intersecting the gates in proper order, thereby remaining close to the input polygon chain. The algorithm finds the approximation with the minimum number of arcs, subject to these constraints.

The main idea for this algorithm is the use of a Voronoi diagram of the tolerance boundary. We have to incrementally maintain one cell in this Voronoi diagram of line segments as we process along the tolerance boundary segment by segment. Recomputing the entire Voronoi diagram in each iteration step would be to expensive, but we can iteratively add n consecutive segments of the tolerance boundary and update the cell in the Voronoi diagram in $O(n)$ total time. Geometric considerations (Lemma 1.8) make the location step in the update easy, leading to constant amortized time per insertion. In total, the algorithm takes $O(n^2 \log n)$ time and $O(n)$ space.

1.1 Problem Setting

We wish to approximate a polygonal chain $P = (p_1, \ldots, p_n)$ by a series of circular arcs (which could include straight line segments, as the limiting case of circles of infinite radius). The endpoints of the arcs are vertices of P. Ideally, we want our approximating curve to have distance at most ε from P. As a first approximation to this problem, one can look at a region formed from strips of width ε centered at the polygon edges. However, in the vicinity of sharp corners, this does not guarantee that the curve remains close to the given points. Figure 1.1 shows a

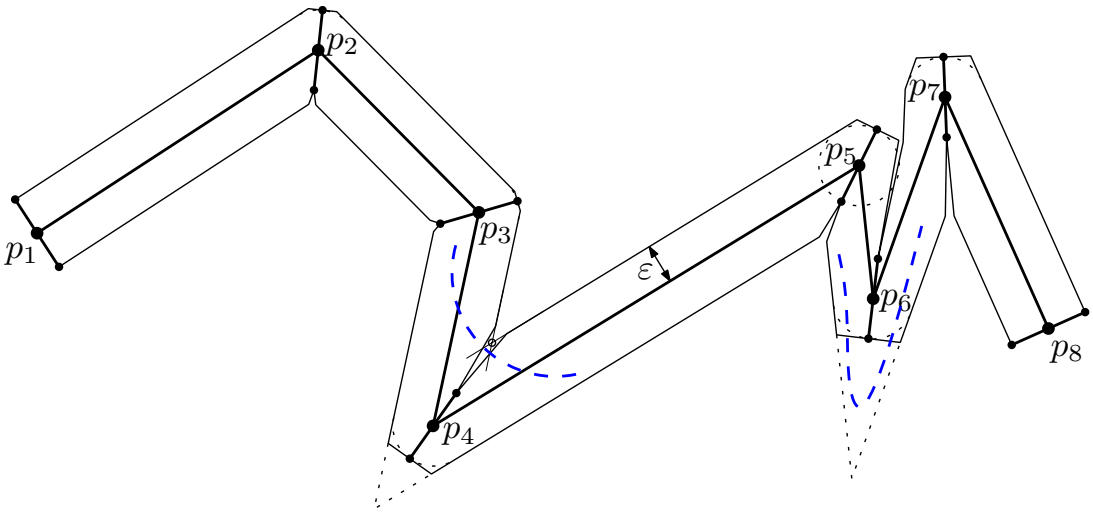


Figure 1.1: Polygonal tolerance region R with gates

circular piece of a hypothetic curve that can shortcut the bend at p_4 if it is only required to remain in the strips. (Also, it might overshoot the bend, as indicated in the vicinity of p_6, although this looks like a theoretical possibility only.) To avoid this, we introduce a *gate* through every vertex. The approximating curve is required to pass through all gates in succession, and the curves are not allowed to pass through a gate twice. This will guarantee that any curve into a point p_i can be joined with any curve out of p_i without danger of an intersection other than at p_i.

For our problem, we assume that we are given a polygonal "tolerance region" R and a sequence of gates $g_1, g_2, \ldots, g_n$, which are segments through the points p_i. Each gate crosses P. We will refer to endpoints of gates lying to the left of P as we walk from p_1 to p_n as left endpoints and the other endpoints as right endpoints. We require that the gates do not cross each other. We require that the input satisfies the following assumptions:

(A) R is a simple polygon passing through all gate endpoints; the boundary of R goes through g_1 and g_n.

(B) R does not intersect the interior of gates or cross the segments connecting corresponding endpoints of successive gates.

(C) No line through two points on successive gates g_i and g_{i+1} crosses the portion of R connecting g_i with g_{i+1}.

(Assumption (B) is actually a consequence of (C).) Ideally, the gate g_i at vertex p_i is a line segment of length 2ε centered at p_i that bisects the angle $p_{i-1}p_ip_{i+1}$. For a convoluted curve with sharp bends close together, we might have to to reduce the width of R in order to fulfill condition A; and we might have to shorten the gates in order to fulfill condition B, as shown in the right part of Figure 1.1 and

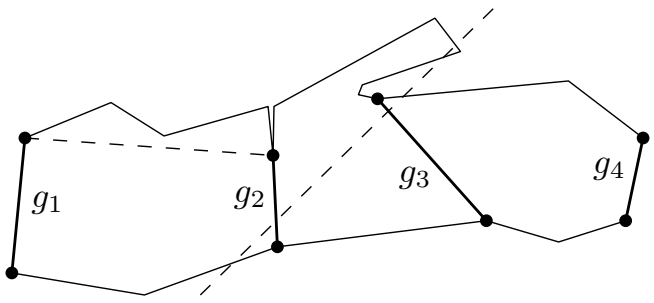


Figure 1.2: Polygonal tolerance region R with gates. Gate g_2 has been "shortened" to fulfill condition B between g_1 and g_2. Condition C is violated between g_2 and g_3.

in the left part of Figure 1.2. In contrast, condition C, beyond what is required for condition B, is likely not an issue in practice: it prevents the boundary of R from making "wild" turns like in the middle of Figure 1.2. The end gates g_1 and g_n partition the boundary of R into a *left* boundary and a *right* boundary. In the illustrations, P will usually be oriented from left to right; then the left boundary is on top and the right boundary is below.

Modeling the curve approximation problem by an appropriate tolerance region with gates is a problem of its own, which we do not treat here. Eibl and Held [38, 47] have methods that can be adapted to produce such gates and tolerance regions. In Figure 1.1, we have chosen to approximate the "ideal" circular boundary at the outer angle of each vertex by a single edge of R. One can use more edges to get a finer approximation, or one could also choose to approximate the circular arc from inside, to get a guaranteed upper distance bound of ε. Our time bounds assume that R has constant complexity between successive gates and thus the total size of R is proportional to n.

Definition 1.1 (proper gate stabbing)**.** *A circular arc stabs gates $g_i, g_{i+1}, \ldots, g_j$ properly, if:*

1. *the circular arc passes through each gate $g_m \in \{g_i, \ldots, g_j\}$ from the side of $\overline{p_{m-1}p_m}$ to the side of $\overline{p_m p_{m+1}}$*
2. *the circle on which the arc lies intersects each gate only once.*

Condition 2 of this definition is necessary for our algorithm, but it excludes arcs that might seem reasonable: an arc from p_i to p_j might intersect each intermediate gate only once, but the continuation of the arc beyond p_j might bend back and intersect, say, g_j and g_{j-1} a second time, see Figure 1.3. This would be a sensible arc, but it is excluded by our definition. But such a situation can only happen if the gates are very close together (relative to their length).

Definition 1.2 (valid circular arc)**.** *A circular arc a_{ij} with starting point p_i and endpoint p_j is a valid arc if:*

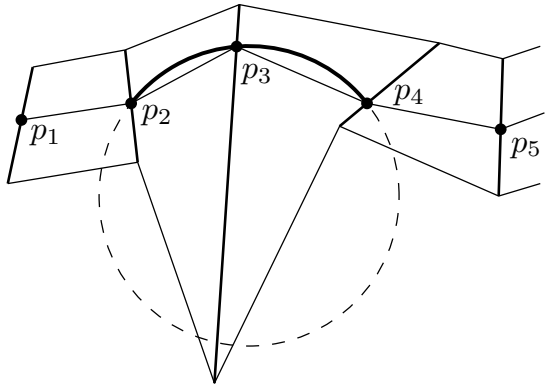


Figure 1.3: A circular arc in a hypothetical tolerance region R that is not valid because it violates Condition 2 of Definition 1.1.

- *the arc stabs the gates $g_{i+1}, \ldots, g_{j-1}$ properly,*
- *the arc does not cross the boundary of the tolerance region R.*
- *the arc reaches p_i from the correct side of g_i and reaches p_j from the correct side of g_j.*

Note that because R passes through the gate endpoints, any arc that goes through a series of gates without crossing the tolerance boundary must go through them in the correct order, so we do not need to test this separately. In contrast to the intermediate gates, we allow the circle on which the arcs lies to intersect g_i and g_j more than once.

We can split the problem of determining if a valid circular arc connects p_i with p_j into three parts. First, we compute the set of all arcs between p_i and p_j that stab all intermediate gates properly (Sect. 1.2). Second, we compute all arcs that start at p_i and end at p_j, reaching both from the correct side (Sect. 1.3). Third, we compute all arcs between p_i and p_j that do not intersect with the tolerance boundary (Sect. 1.4). A valid circular arc has to be a member of all three result sets.

1.2 Stabbing the Gates

Given a point p and a gate g, denote by b_l the bisector of p and g's left endpoint, and by b_r the bisector of p and g's right endpoint.

Lemma 1.1. *The centers of all circles passing through a vertex p_i and intersecting a gate g_j exactly once lie in a double-wedge whose boundary is b_l and b_r. Specifically, they lie in the parts of the double wedge where one of the half planes bounded by b_l and b_r includes p_i and the other excludes it. (Figure 1.4 illustrates this.) In the degenerate case where b_l is parallel to b_r the region containing all centers is the strip between the bisectors.*

Proof. Consider the intersection of the half plane bounded by b_l that includes p_i and the half plane bounded by b_r that excludes p_i. Points in the interior of this region are closer to p_i than the right endpoint of the gate and are also closer to the left endpoint than to p_i. Disks centered in this region which have p_i on their boundary include the left endpoint and exclude the right endpoint of the gate. Therefore all circles centered in the wedge intersect the gate exactly once. The case for the second wedge is symmetric. This argument works for the degenerate case, also, but in this case all circles will include the nearer gate endpoint and exclude the further one.

Centers of circles that are located in the same region as p_i outside of the double-wedge are always closer to p_i than to the endpoints of the gate. Therefore these circles exclude the endpoints if they pass through p_i. These circles can not intersect the gate only once, unless the circle is tangent to the gate. Looking at the other side of the double-wedge boundary, all centers of circles located here are closer to the endpoints of the gate than to p_i. Each disk which includes p_i has to include the endpoints and its boundary does not intersect the gate at all. □

Lemma 1.2. *The region of the centers of all circles passing through a vertex p which are tangent to the gate g or intersect it twice forms a parabolic region (in Figure 1.4 the parabolic region is the filled region to the left of the double-wedge). The boundary of the parabolic region is given by a parabolic piece, defined by the centers of the circles which are tangent to the gate, and by two pieces of the boundary double-wedge. In the degenerate case when the bisectors are parallel the parabolic region is empty.*

Proof. Geometric analysis proves the claim. To find the centers of circle which are tangent to the gate, we are looking for points which are equidistant from a point (p) and a line (g), which is the geometric definition of a parabola. Let c_l be intersection point of the parabola and the bisector b_l. Then c_l is the center of a circle which is tangent to g at the left boundary point of g. Symmetrically the intersection point c_r of the parabola with the bisector b_r is the center of a circle which is tangent to g at the right boundary point. These intersection points are two corners of the parabolic region. The intersection point c_m of the bisectors b_l and b_r is the center of the circle which passes through p and both boundary points of the gate. This is last corner of the parabolic region. The edge connecting c_l, c_m corresponds to points which are equidistant from the left boundary point and p and are closer to the gate than the points on the parabola. Therefore circles centered at these points pass through the left boundary point and have to intersect the gate in one additional point. Symmetrically the circles centered on the edge c_m, c_r pass through the right boundary point and also intersect the gate in one additional point. So all the points in the interior of the parabolic region are centers of circles which intersect the gate twice.

In the degenerate case p and the left and right endpoints of g are collinear, so no circle is tangent to g or intersects it twice.

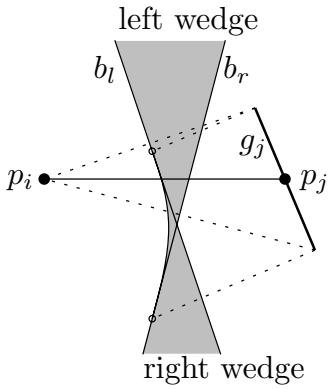


Figure 1.4: The shaded area is the region of all centers of circles passing through p_i and gate g_j. The circles with centers close to the intersection of b_l and b_r, in the region with the curved boundary, intersect g_j twice and are not considered as centers of valid arcs.

□

By Definition 1.1, an arc stabs the gates properly only if every gate is intersected only once. Therefore the centers of circular arcs stabbing an intermediate gate are located in the double wedge of the gate. For the first and last gates of the arc we insist that the arc goes through the original point located at the gate. Thus the first and last gates are treated differently from the intermediate gates (see Subsection 1.3).

According to Lemma 1.1, one wedge is the region of the centers of disks including the left endpoint of the gate and excluding the right endpoint. Circular arcs centered in this region pass the gate from the correct side, according to the stabbing condition, if they are in CCW (counter-clockwise) orientation. In CW (clockwise) orientation, the arc would walk around the left endpoint before intersecting the gate. The unbounded part of this wedge lies to the left of P. Symmetrically the circular arcs in the other wedge need CW orientation to pass the gate in the correct direction, and the unbounded part of this wedge lies to the right of P.

So from now on we talk about the left wedge and the right wedge. A circular arc stabbing through the gates cannot change its orientation.

Lemma 1.3. *A circular arc α starting at a point p stabs gates $g_i, \ldots, g_j$ properly if and only if its center lies in the intersection of the left wedges or the intersection of the right wedges defined by p and the gates.*

Proof. Any point in the intersection of the left wedges of all the gates is the center of an arc that passes through p and all of the gates in a CCW direction. Any CCW arc that passes through all of the gates will have its center in each left wedge, so the intersection of these wedges will include it. The argument for right wedges is symmetric.

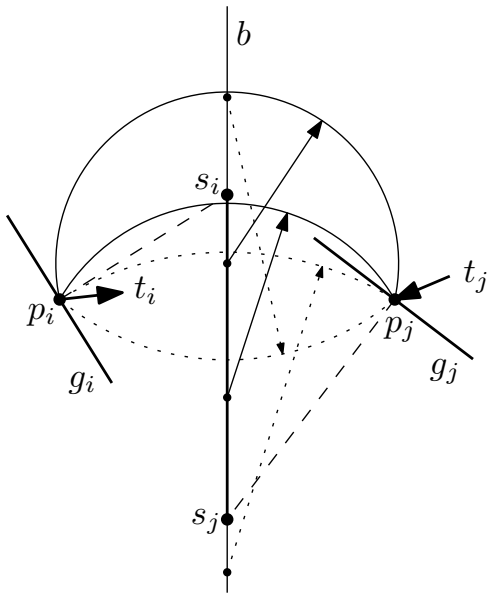


Figure 1.5: Illustration for Lemma 1.5. In this example, the centers of valid CW arcs form the line segment $s_i s_j$. There are no valid CCW arcs. A few representative candidate arcs are shown.

□

Computing the intersection of the left wedges and symmetrically of the right wedges, corresponds, according to the proof of Lemma 1.1, to computing the intersection of half planes. So the intersection of n wedges is the intersection of $2n$ half planes. This gives us the following bound:

Lemma 1.4. *Incrementally computing the two regions of centers of all valid circular arcs passing through a point p_i and stabbing all gates $g_i, g_{i+1}, g_{i+2} \ldots, g_j$ properly, for $j = i+1, \ldots, n$, can be done in $O(n \log n)$ time and $O(n)$ space.*

Proof. It is the incremental intersection of $O(n)$ half-planes.

□

1.3 Arc Endpoints

All arcs that start at p_i and end at p_j have their centers on the bisector of the segment connecting p_i and p_j. Since a valid circular arc from p_i to p_j must reach each endpoint from the correct side of its gate, we know for each circle whether the arc from p_i to p_j must go in the CW or in the CCW direction, or if none of the arcs is valid. (When the circle is tangent to both gates, both directions are possible.) Straightforward geometric arguments lead to the following characterization of the desired arcs, see Figure 1.5.

Lemma 1.5. *Let b be the perpendicular bisector of the segment between p_i and p_j. Let s_i be the point of b which is the center of a circle tangent to g_i at p_i, and let s_j be defined symmetrically. The centers of all CW arcs that reach both p_i and p_j from the correct side lie in the intersection of two rays that are subsets of b. One has s_i as its endpoint and the other has s_j as its endpoint. The same is true for CCW arcs.*

Proof. This ray is the intersection of the CW rays for both endpoints of the arc. □

1.4 Staying within the Tolerance Boundary

The tolerance boundary R consists of two polygonal chains, one on each side of the original polygonal chain P. For a CW arc we will only check that it does not cross the boundary on the *left* side of P. It cannot cross the boundary on the right side of P if it passes through all gates, by assumption (B), and therefore we need not check for such an intersection explicitly. (For a CCW arc, the situation is symmetric.)

A circle passing through point p does not intersect or contain any edge on a polygonal chain C if its center lies closer to p than to any point on C. That is, if we compute the Voronoi diagram of $C \cup p$, the center of the circle must lie in point p's region, $V(p)$.

This is not quite the condition that we want, namely that a circular arc does not cross chain C. The Voronoi region guarantees that an entire circle does not cross C. However, in our case these are equivalent.

Lemma 1.6. *If an arc from g_i to g_j does not intersect a tolerance boundary between g_i and g_j then neither does the circle on which that arc lies.*

Proof. Look at the arc between consecutive gates g_k and g_{k+1}. Let q and q' be the intersection points with these gates. By assumption (C), the line ℓ through q and q' does not intersect the tolerance boundary between g_k and g_{k+1}, i.e., the tolerance boundary lies entirely on one side of ℓ. For a CW arc, the tolerance boundary in question lies on the left side of ℓ. On the other hand, ℓ is the line that splits the circle into the arc from q to q' (on the left side) and into the opposite part which is not used. Thus the part of the circle which is not used can never intersect the relevant part of the tolerance boundary.

□

While we could compute the entire Voronoi diagram of $C \cup p$ to determine $V(p)$, this would be too expensive. Fortunately, we can iteratively add n consecutive segments of C and update p's Voronoi region $V(p)$ in $O(n)$ total time.

As shown by Drysdale [32], Voronoi regions are "generalized star shaped". This means that a shortest segment from a boundary point to a nearest point in the shape defining the region lies entirely within the region.

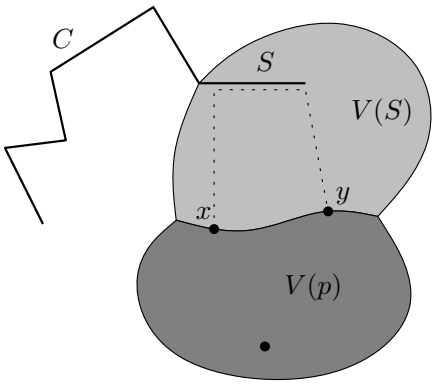


Figure 1.6: Schematic illustration for Lemma 1.7.

Lemma 1.7. *Each segment added will either cause no change to $V(p)$ or will replace a section of $V(p)$ by at most three new segments (two straight lines and a parabola). (If $V(p)$ is unbounded we think of an edge "at infinity" connecting the two infinite rays, so that these three "segments" are considered consecutive.)*

Proof. Suppose we add a new segment S to the end of C. First we show that the added pieces on the boundary of $V(p)$ are connected. Let x and y be two points on the boundary between $V(p)$ and $V(S)$ (x and y can also be chosen "at infinity"). Draw shortest segments (or rays for the piece "at infinity") from x and y to S. Because Voronoi regions are generalized star shaped, both of these segments lie within $V(S)$ and cannot be crossed by another Voronoi region. S itself cannot be crossed by another Voronoi region. There is a closed curve formed by a part of S, the two segments, and the boundary of $V(p)$ between x and y, cutting the plane into two parts, see Figure 1.6. Since S is the (current) last segment of C, one of these parts contains no other segments of C. It follows that the corresponding part on the boundary of $V(p)$ belongs completely to $V(S)$, establishing a connection between x and y.

The Voronoi bisector between a point p and a segment S is formed by 2 straight rays and a parabolic arc. The new parts on the boundary of $V(p)$ must be a part of this bisector.

□

There are two parts to updating p's Voronoi region $V(p)$ when adding a segment S to the diagram. First, we find a place on the boundary of $V(p)$ that is equidistant from p and S, if such a place exists. If so, we walk around the boundary of $V(p)$, eliminating boundary sections until we reach the other place on the boundary where p is equidistant from S. (Note that either of these places could be "at infinity".)

The second part is easy — walk around the boundary of $V(p)$ from the starting point, eliminating obsolete bisector segments until you get to the finish point.

Because C is a polygonal chain, the first part is also easy. $V(p)$ is bounded by bisector pieces between p and a subset of the segments in C. Of the segments in this subset, there is a first segment F and a last segment L, according to the order along the chain.

Lemma 1.8. *If $V(p)$ changes, then its boundary with either $V(F)$ or $V(L)$ must change.*

Proof. The intuition is, if you can't go through the chain C, then the only way to get to $V(p)$ is through $V(F)$ or $V(L)$.

If the chain from F to L consists of only F and L (which could be the same segment), the lemma is trivially true. Otherwise consider the union of the chain C between F and L exclusive, the boundary of $V(F)$ from the endpoint it shares with the next segment on C to the end of its boundary with $V(p)$, and the boundary of $V(L)$ from the endpoint it shares with the segment before it on C until the end of its boundary with $V(p)$. If $V(p)$ is bounded these two boundaries end at the same point — the point where $V(p)$, $V(F)$, and $V(L)$ meet. If $V(p)$ is unbounded then its boundaries with $V(F)$ and $V(L)$ end in infinite rays. In either case, this union separates the plane into two parts, one including p (the inside) and the other not including p (the outside). We will call this union the separator. Note that F and L are defined to lie outside of this separator (except for the endpoint that is part of the separator).

Suppose that a segment S is added that changes $V(p)$. The previous segment on C is either L or some segment that did not modify $V(p)$. In either case, the endpoint shared with that previous segment is outside of the separator, so we know that at least part of S lies outside of separator.

If S crosses the separator, then it cannot cross C, because the chain is simple. If it crosses the Voronoi boundary of $V(F)$ then the part of the boundary between the crossing point and the end of the boundary between $V(p)$ and $V(F)$ will be eliminated. A similar argument holds for L. Thus if S crosses the separator then the boundary of $V(p)$ with either $V(F)$ or $V(L)$ must change.

If S does not cross the separator, pick some point q that is on the boundary of the new $V(p)$ that was not on the boundary of the old $V(p)$ and let E be the shortest segment from q to a point on S. E must lie entirely in $V(S)$ and must cross the separator. It cannot cross C. The rest of the analysis is exactly as in the paragraph above, with E replacing S.

□

Lemma 1.9. *For a fixed gate g_i, we can incrementally compute the regions of centers of all circular arcs that pass between g_i and each gate g_j, without crossing the tolerance boundary, for $j = i+1, i+2, \ldots, n$, in $O(n)$ time and space.*

Proof. Incrementally add segments from C and amortize the update time. We have shown that the centers of CW [CCW] arcs are the region of $V(p_i)$ in the

Voronoi diagram of p along with the CW [CCW] boundary between g_i and g_j. We can compute these regions incrementally. It takes constant time to test if segment S changes the boundary between p and either F or L, so the total time for finding starting points is $O(n)$.

Walking along the boundary of $V(p)$ will take time proportional to the number of pieces eliminated. Because an eliminated piece is removed and never reappears, the total time for this step in all n insertions is bounded by the number of boundary pieces added. This is at most $3n$, because a bisector curve between p and a segment consists of at most three pieces. Thus this requires time $O(n)$.

□

1.5 Computing the Shortest Path

To determine the approximation with the minimum number of arcs we look at the directed acyclic graph of all possible valid arcs and find the shortest path from p_1 to p_n. The following theorem summarizes how to find the valid arcs from p_i to p_j.

Theorem 1.1. *A point c is the center of a valid CW circular arc from p_i to p_j if and only if it is in the intersection of*:

- *the intersection of the right wedges between p_i and each of the gates g_{i+1} through g_{j-1};*
- *the region of $V(p_i)$ in the Voronoi diagram of p_i and all of the segments on the left boundary between g_i and g_j; and*
- *all points in the intersection of two rays contained in b, one with endpoint s_i and the other with endpoint s_j, where b, s_i, and s_j are as defined in Lemma 1.5.*

The conditions for valid CCW arcs are symmetric.

Proof. Direct consequence of earlier lemmas.

The first condition guarantees that the arc passes through gates g_{i+1} through g_{j-1}. The second guarantees that it does not intersect the tolerance boundary. The final condition guarantees that it has p_i and p_j as endpoints and approaches each from the correct side.

□

We find the possible arcs from a point p_i to all points further along P incrementally. We maintain the intersection of the right wedges, the intersection of the left wedges, the Voronoi region of p_i with the left boundary, and the Voronoi region of p_i with the right boundary. At each step we update each of the four items. We intersect each bisector ray with an intersection of wedges and

with a Voronoi region, and then test if the intersections overlap. Because wedge intersections and $V(p_i)$ are convex these intersections require $O(\log n)$ time.

Note that we can quit early as soon as both wedge intersection regions become empty. This may lead to a better behavior of the algorithm in practice than the worst-case time bound proved in the theorem below.

Theorem 1.2. *Given an open polygonal curve $P = (p_1, \ldots, p_n)$, a polygonal tolerance boundary of size $O(n)$, and a gate for each p_i, we can approximate P by a minimum number of valid circular arcs in $O(n^2 \log n)$ time and $O(n)$ space.*

Proof. For each starting point p_i we can determine the points p_j $(j > i)$ that can be reached by a valid arc in $O(n \log n)$ time and $O(n)$ space. In the shortest path algorithm, it is sufficient to scan the outgoing arcs of p_1, p_2, p_3, and so on, in succession. Therefore, once the valid arcs out of p_i are scanned, they need not be stored any longer, and hence the algorithm needs only $O(n)$ space.

□

For the min-# problem for *polygonal* approximation the best known running time is $O(n^2 \log n)$ for three out of the four common error criteria , see the results of Eu and Toussaint [39] as well as Imai and Iri [51, 52]. Our algorithm solves this problem with curves of higher order with the same time complexity. The error criteria refer to the way the error of an approximating segment is measured. Only for the E1 error criterion (the maximum distance between the approximating segment and the vertices of the original polygonal curve that lie between start and endpoint of the segment) there is an algorithm with a faster running time of $O(n^2)$ presented by Imai and Iri in [52].

Remark. The algorithm can be extended to optimize other criteria than the number of arcs, e.g. the arc length, or some weighted mixture of criteria. When the interval of possible centers of valid arcs from p_i to p_j has been determined, one must be able to pick the best one of them and compute its "weight", which is used for the shortest path calculation.

As Held mentioned in [47] conventional biarc algorithms (also used in industry) operate on discrete sets of points (and tangent vectors), by fitting biarcs between selected pairs of points. Therefore the restrictions of our algorithms are common. Nevertheless, we are aware that certain restrictions of the solution are not completely natural. In particular, one might allow arcs and biarcs that do not start and end at original points. Using these restrictions simplifies the problems, and we do not know to solve them otherwise.

The results on arc approximations have been presented at the 22nd European Workshop on Computational Geometry (EWCG) in Delphi, in March 2006 [33].

Chapter 2

Approximation with a Minimum Number of Biarcs

The sequence of arcs produced in the previous algorithm may have arbitrary corners at the vertices. In many situations, a smooth curve is desired. We now assume that an oriented tangent direction is specified for each vertex of the open polygonal curve. (If such tangent information is not available, it can also be computed from the point data alone, using various tangent estimation methods.)

Our algorithm will select a subsequence of the input points and interpolate between them smoothly by *biarcs*, pieces consisting of pairs of circular arcs, respecting the tangent directions at the points which are used. Our algorithm will find such an approximation with the minimum number of biarcs given a set of gates and a tolerance region in $O(n^2 \log n)$ time and $O(n^2)$ space.

2.1 Problem Setting

We wish to approximate a polygonal chain $P = (p_1, \ldots, p_n)$ with given oriented tangent direction t_i for every vertex p_i by a series of biarcs. The endpoints of the biarcs are vertices of P. Like for the approximation with circular arcs, we assume that we are given a polygonal "tolerance region" R and a sequence of gates $g_1, g_2, \ldots, g_n$, which are segments through the points p_i. Each gate crosses P. In this setting the gates would ideally be perpendicular to the tangent directions t_i, but we do not require this. We require that the input satisfies the following assumptions:

(A′) R is a simple polygon passing through all gate endpoints;

(B′) R does not intersect the polygon or the interiors of the gates.

(C′) Each tangent t_i passes through gate g_i in the same direction as the original polygonal chain P; that is, from the side of the gate on which $p_{i-1}p_i$ lies to the side on which p_ip_{i+1} lies.

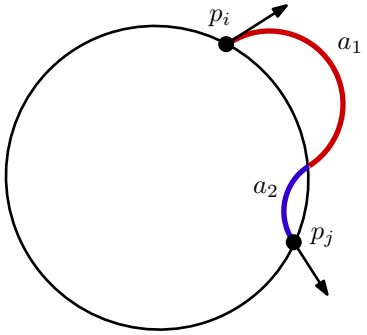


Figure 2.1: The joint circle, and an S-shaped biarc with both tangents pointing outside the joint circle

Again, we first find all valid biarcs and then build the directed graph of these biarcs from the start point to the end point of the polygonal curve. The last step is the computation of the shortest path as in the previous chapter. The main difference between the two algorithms is the computation of the valid arcs/biarcs.

2.2 Biarcs

Biarc curves were introduced by Bolton [16] and are used for curve approximation in a tangent-continuous manner. A biarc consists of two circular arcs that share an endpoint with a common tangent. This common endpoint is called the *joint* of the biarc. Given two points p_i and p_j with two tangent vectors t_i, t_j at these points, a biarc B_{ij} between p_i and p_j is characterized in the following way [16, 47]:

- B_{ij} consists of two consecutive circular arcs, a_1, a_2
- a_1 is an oriented arc from p_i to point p_{joint} and a_2 is an oriented arc from p_{joint} to p_j;
- a_1 matches the tangent vector t_i at the point p_i and a_2 matches the tangent vector t_j at p_j;
- both arcs have a common tangent at p_{joint}.

These conditions leave one degree of freedom.

2.2.1 Joint Circle

The locus of possible joints forms a circle J that passes through p_i and p_j [21, 70, 78], see Figure 2.1. For each point on this *joint circle* J, there is a unique biarc which uses this point as the joint. There are some degenerate cases: as a limiting case, the joint could be one of the points p_i or p_j; the joint circle might be a line; if there is a circle through p_i and p_j with the given tangents, this is the joint circle, but all joints on this circle lead essentially to the same biarc.

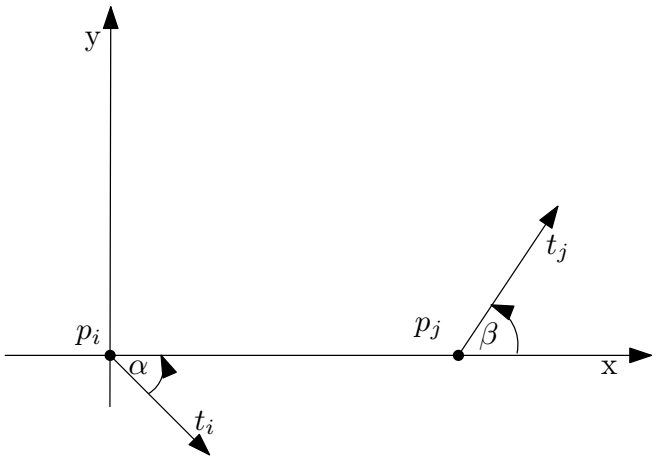


Figure 2.2: Moving the point-tangent pairs p_i, t_i and p_j, t_j to a local coordinate system.

After moving the biarc to a local coordinate system, so that p_i is the origin and the vector $\vec{p_ip_j}$ is the positive direction of the x-axis, the center point $C_J(c_x, c_j)$ of the joint circle is defined as follows (see Young et al. [79]):
Let α be the CCW angle from t_i to $\vec{p_ip_j}$ and β the CCW angle from $\vec{p_ip_j}$ to t_j, as shown in Figure 2.2. The center of the joint circle $C_J(c_x, c_j)$ is:

$$c_x = \frac{\|p_ip_j\|}{2} \tag{2.1}$$

$$c_y = \frac{-\|p_ip_j\|\cos\theta}{2\sin\theta}, \text{ with } \theta = \frac{-\alpha-\beta}{2} \text{ and } \theta \in (-\pi, \pi)\,. \tag{2.2}$$

The unsigned radius of the joint circle is:

$$|R| = \frac{\|p_ip_j\|}{2\,|\sin\theta|}. \tag{2.3}$$

For implementation issues it is important to distinguish between the joint circle and the degenerate cases of a segment or just a point as locus of the joints. Therefore we present how to distinguish and construct them with a few geometric routines. The case distinction is based on the orientation of the tangents t_i,t_j and their relative positions.

- The tangents t_i and t_j are parallel with the same orientation:
  The locus of the joints is the segment $\overline{p_ip_j}$, see Figure 2.3.

- The tangents t_i and t_j are parallel with opposite orientation:
  The locus of the joints is the circle through p_i and p_j centered at the midpoint of the segment $\overline{p_ip_j}$,see Figure 2.4.

- The tangents t_i and t_j are the same :
  The locus of the joints is a point, either p_i or p_j.

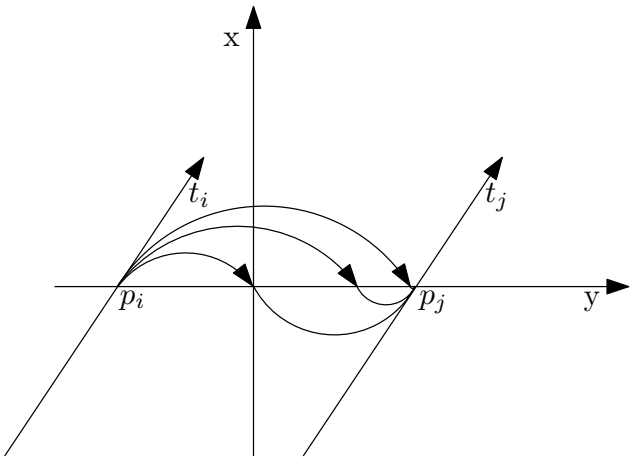


Figure 2.3: The segment $\overline{p_i p_j}$ is the locus of all joints.

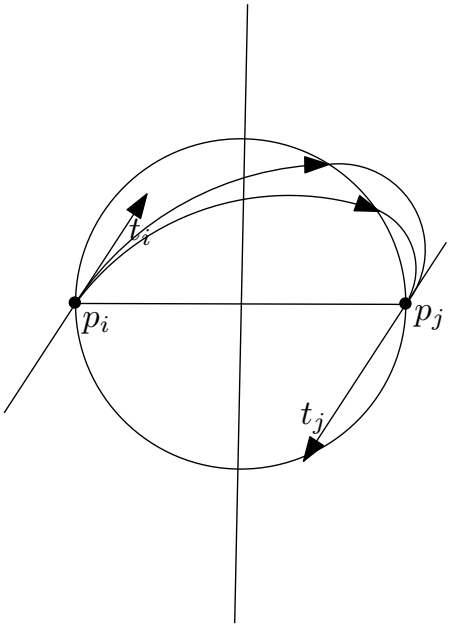


Figure 2.4: The center of the joint circle is the midpoint of the segment $\overline{p_i p_j}$.

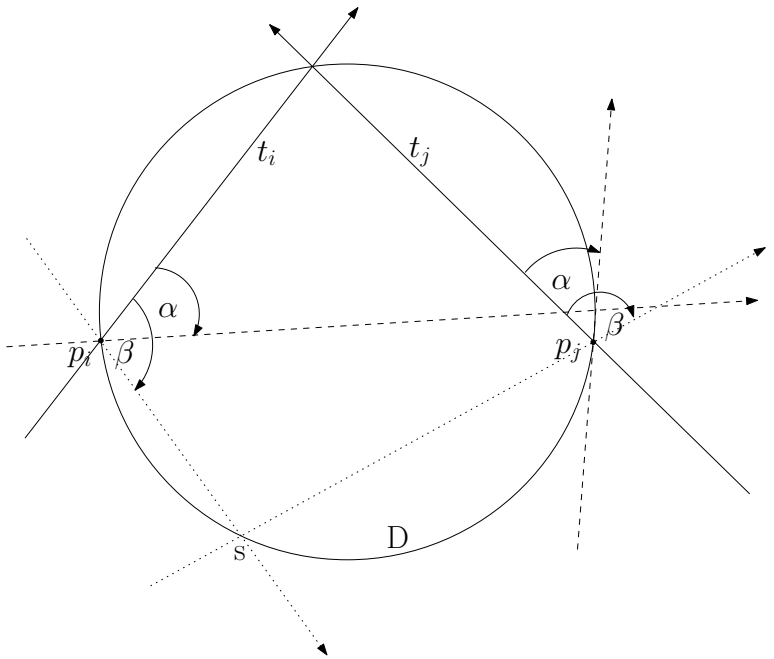


Figure 2.5: Rotating the tangents t_i, t_j around the points p_i, p_j at the same pace, the intersection points of the tangents are located on a circle D. Let s be any point on D. The tangent t_i and the ray from p_i to s and the tangent t_j and the ray from s to p_j form the same angle.

- The tangents t_i and t_j are neither parallel nor the same:
  First construct the circle which is defined through the point triple p_i, p_j and the intersection point of the two tangents t_i, t_j. Let s be any point on this circle D. The angles between t_i and the ray from p_i to s and t_j and the ray from s to p_j are by construction the same, see Figure 2.5. Next intersect D with the bisector of the segment $\overline{p_i p_j}$. The two intersection points correspond to two possible centers for the joint circle, but only one of these centers is feasible. Looking at the two tangents t_i, t_j, only one of the two intersection points is on the same side (regarding the tangent directions) of the two tangent lines t_i and t_j. This point is the center of the joint circle, see also Figure 2.6.

The circle with the second intersection point as center forms only equal angles with the tangent lines if the orientation of one of the tangents is flipped, see also Figure 2.7. Therefore the construction of the joint circle can be geometrically done with a few simple routines: first the construction of a circle through a point triple, then construction of the bisector of a segment, next the intersection of circle and bisector, last check which intersection point lies on the same side of two given lines.

We will ignore in the sequel the degenerate cases and will only refer to the joint circle.

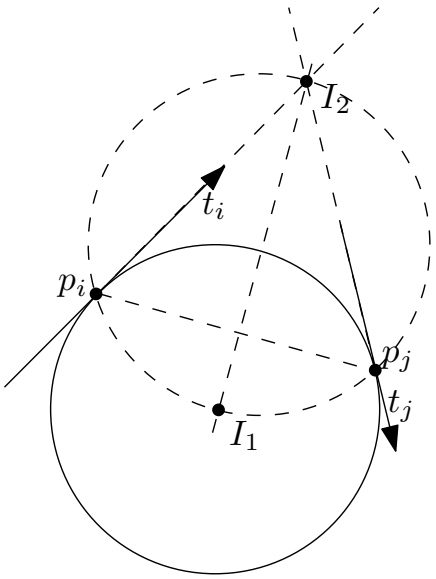


Figure 2.6: Construction of the joint circle defined through the two point-tangent pairs p_i, t_i and p_j, t_j. The intersection point I_1 is to the right of t_i and to the right of t_j and therefore the center of the joint circle. In this example the joint circle is a circle through p_i and p_j with the given tangents, but all joints on this circle lead essentially to the same arc as biarc.

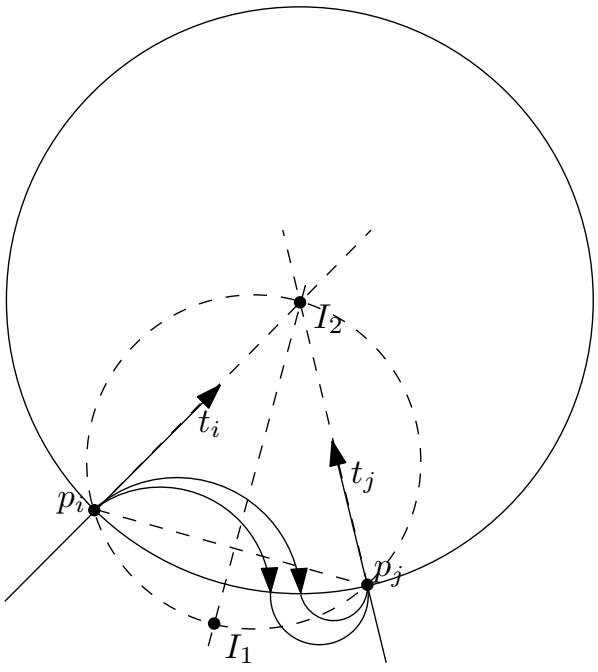


Figure 2.7: Construction of the joint circle with the point-tangent pairs like in Figure 2.6 but the orientation of the tangent t_j is flipped.The intersection point I_1 is to the right of t_i but to the left of t_j and therefore the center of the joint circle is I_2.

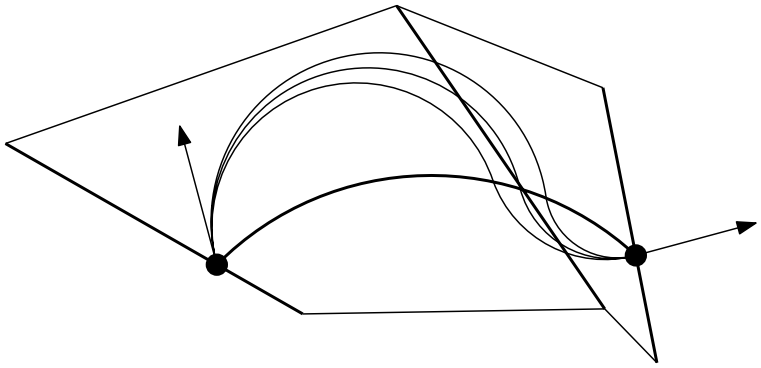


Figure 2.8: We allow a biarc to intersect an intermediate gate three times.

Proposition 2.1. *One circular arc of the biarc lies outside the joint circle J, and the other lies inside J, except for their endpoints, which lie on J. Both tangents t_i and t_j point to the same side (either inside or outside) of J, and they form equal angles with J. (In fact, the last property characterizes the joint circle.)* □

2.2.2 Valid Biarcs

Definition 2.1 (Valid biarc)**.** *A* valid biarc B_{ij} *from p_i to p_j consists of two circular arcs a_1 and a_2 and satisfies the following conditions*:

- *a_1 matches t_i at the point p_i, a_2 matches t_j at p_j, and they meet at a point on the joint circle.*
- *B_{ij} stays inside the tolerance boundary.*
- *B_{ij} intersects the gates g_i and g_j only in p_i and p_j.*

The joint, which is the ending point of a_1 and the starting point of a_2, is not required to be an original point. The joint must of course lie inside the tolerance region. Note that in comparison to the arc approximation of Definition 1.2, we have relaxed the gate stabbing condition. The arcs a_1 and a_2 are allowed to intersect the gates of the starting and ending points only once, but intermediate gates can be intersected more than once. Forbidding these multiple intersections would mean that, in a family of biarcs with the same endpoints, some biarcs that lie between permitted biarcs might be excluded, which is not natural. See Figure 2.8. The restrictions on intersecting g_i and g_j guarantee that successive biarcs will not intersect except at endpoints.

2.3 Circular Visibility Regions

For each possible starting point p_i of a biarc, the tangent direction t_i is fixed. The pencil of circular arcs starting in this direction forms a *circular visibility region*

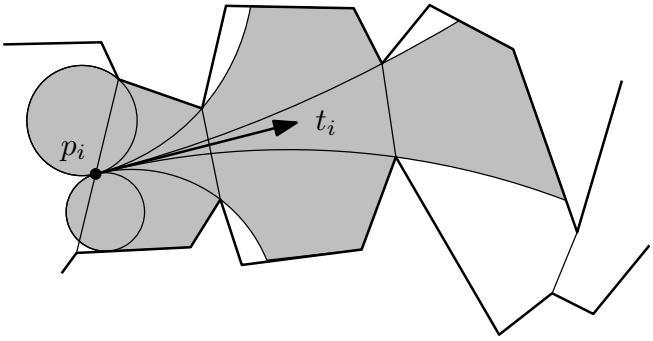


Figure 2.9: A circular visibility region W_i

W_i inside the feasible region R, see Fig. 2.9. The arcs forming W_i terminate when they reach p_i; since we want to construct valid biarcs, we are not interested in arcs that intersect p_i a second time.

To find a valid biarc that starts at p_i and ends at p_j we need to reach a point on the joint circle J via a valid arc from p_i and then continue via a valid arc to p_j. The possible joints from the perspective of p_i are the intersection of J and the circular visibility region of p_i. By reversing the direction of the second arc and tangent we can compute the second arc in the same way. We will use arc $\tilde{a}_2$, which has opposite orientation and whose tangent at p_j is $\tilde{t}_j$, the reverse of t_j. We will call this circular visibility region $\tilde{W}_j$. Our goal is to find all points on J which are in both circular visiblilty regions W_i and $\tilde{W}_j$.

As a first step in this process we determine the portion of each gate that is within W_i for each point p_i and the portion of each gate that is in $\tilde{W}_j$ for each point p_j. These portions consist of at most three intervals and can be stored in $O(n^2)$ space (see Lemma 2.1). In the second step we check the existence of a valid biarc between every pair of vertices p_i and p_j. For each pair p_ip_j, we will identify an interval $g_{l-1}, \ldots, g_{r+1}$ of gates where the joint is restricted to lie. In this interval, the joint circle is not intersected by the boundary of R. This makes it easy to test for the existence of a valid joint. This step uses the pre-computed information about the intersection of circular visibility regions with gates.

We could compute the intersection of W_i with all later gates by using intersections of wedges and Voronoi regions, as we did in chapter 1. However, because we know the tangent at p_i we can do this more efficiently by computing W_i directly. The pencil of circular arcs consists of an interval of possible curvatures. As we proceed from gate to gate and walk along the left and right tolerance boundaries, the interval of curvatures either remains unchanged or shrinks, but it always remains a single interval.

Lemma 2.1. *For a given point p_i, the oriented circular visibility regions of W_i and their intersection with all gates can be computed in $O(n)$ time. The part of a gate that is visible in W_i consists of at most three intervals: one interval where the gate is reachable in the forward direction, and two segments where the gate*

is reachable in the backward direction by clockwise and counter-clockwise arcs, respectively. These intervals may be adjacent.

Proof. We cut each oriented visibility region into two pieces, forward and backward visibility. The forward visibility region is the part of the visibility region which is reached by portions of arcs that do not intersect any gate twice. The backward visibility is the part reached by portions of arcs after they have intersected a gate twice, so they are moving backwards through the gates.

We walk along the left and right boundaries of the tolerance region, determining the intersection between each boundary and the pencil of arcs, and in this way compute the forward visible region. When the last reachable gate is known, we can compute for each gate moving backwards the arcs that build the backward visibility region. The backward part of the visibility region for a gate g_i consists of the arcs that intersect gate g_{i+1} a second time (possibly after passing through even higher-numbered gates twice) and reach back to g_i, plus the arcs that don't cross g_{i+1} but intersect the gate g_i a second time. These arcs correspond to a connected piece of the pencil of arcs and we need to determine the intersection of this pencil part with the corresponding boundary of the tolerance region moving from g_{i+1} to g_i. Because the complexity of the tolerance boundary between two gates is constant we can do the forward and backward visibility computations between two gates in constant time, so the total time required is $O(n)$. The intervals on the gates can be stored for all point pairs in $O(n^2)$ space.

□

Note that the interval on a gate reachable by forward portions of arcs is disjoint from the interval reachable by backward portions of arcs, because a given point is reachable by exactly one arc leaving p_i with tangent t_i. These regions (if non-empty) may join at the point where an arc is tangent to the gate, but if this arc is invalid (because it intersects the boundary) the regions will be separated. See Fig. 2.10.

2.4 Computing Valid Biarcs

We now look at a fixed pair p_ip_j test for a valid biarc between p_i and p_j. The tangent directions t_i and t_j define a joint circle J. For the rest of the chapter we will deal with the situation that the first arc starting at p_i is outside the joint circle J and the second arc is inside. The other case is symmetic.

Each gate $g_{i+1}, \ldots, g_{j-1}$ may or may not fulfill the following conditions:

Condition (Out)

The visibility region W_i from p_i intersects the gate OUTSIDE the joint circle.

Condition (In)

The visibility region $\tilde{W}_j$ from p_j intersects the gate INSIDE the joint circle.

We can test the conditions (In) and (Out) in constant time for every gate.

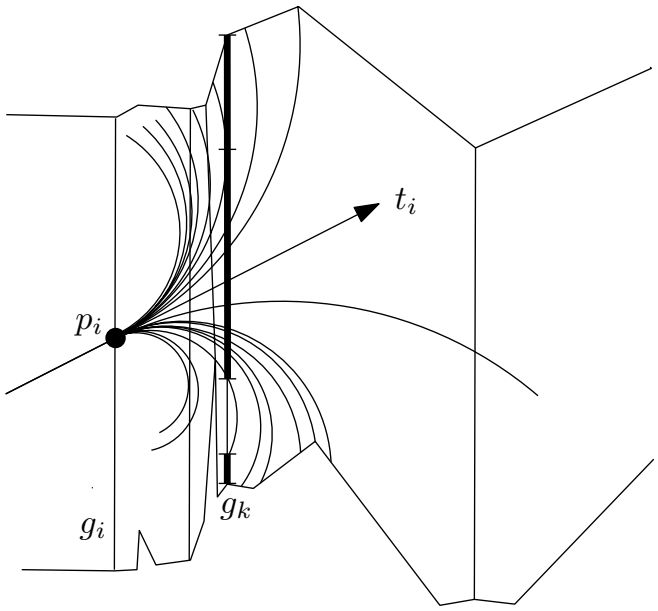


Figure 2.10: Forward and backward visibility segments of region W_i on gate g_k. The intervals of forward visibility and counter-clockwise backward visibility are adjacent.

In the following, we will refer to the region bounded by two successive gates g_{k-1} and g_k and the boundary of R, as the *cell* between these gates, or simply the *cell* g_{k-1}, g_k.

Lemma 2.2. *(a) If the joint circle contains a joint point for a valid biarc in the cell between g_{k-1} and g_k then g_{k-1} satisfies* (Out)*, and g_k satisfies* (In)*.*

(b) If g_k satisfies (In) *then so does $g_{k+1}, \ldots, g_{j-1}$. If g_k satisfies* (Out) *then so does $g_{k-1}, \ldots, g_{i+1}$.*

Proof. We prove only the statements regarding (Out). The arcs starting at p_i in direction t_i start outside the joint circle J. If such an arc enters the joint circle, it remains inside until it returns to p_i (see figure 2.11). Thus, if an arc has reached g_k outside J, the initial part must have passed through $g_{i+1}, \ldots, g_{k-1}$ outside J. This establishes part (b) of the lemma. The same argument works for an arc that reaches J in the cell between g_{k-1} and g_k (part (a) of the lemma).

□

It follows that the sequence $g_{i+1}, \ldots, g_{j-1}$ can be partitioned into three consecutive parts:

(a) an initial part $g_{i+1}, \ldots, g_{l-1}$ (possibly empty) satisfying (Out) but not (In);

(b) a middle part $g_l, \ldots, g_r$, which is either

(b1) a nonempty sequence satisfying neither (In) nor (Out);

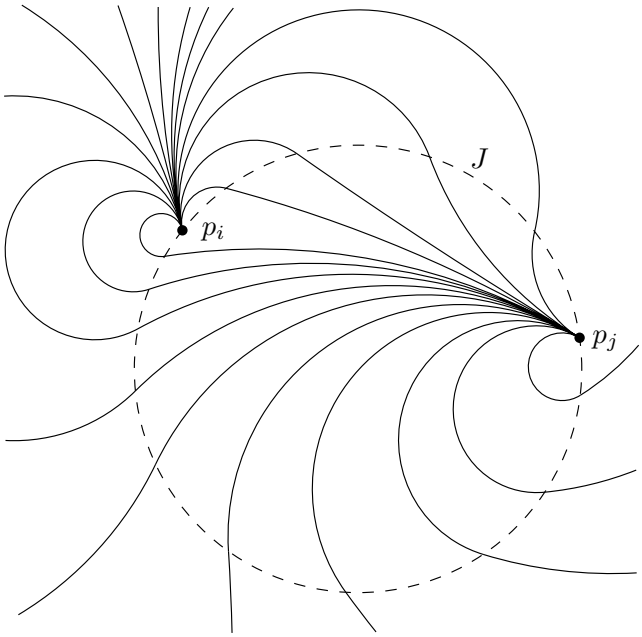


Figure 2.11: A family of biarcs from p_i to p_j and its joint circle J

(b2) a possibly empty sequence satisfying both (In) and (Out);

(c) a final part $g_{r+1}, \ldots, g_{j-1}$ (possibly empty) satisfying (In) but not (Out).

Since the conditions (In) and (Out) can be tested in constant time, the positions l and r can be identified by binary search in $O(\log n)$ time. From Lemma 2.2a it is clear that in case (b1), there can be no valid biarc, and in case (b2), the joint must be in the cells between g_{l-1} and g_{r+1}.

Let us now concentrate on case (b2): We treat the cells g_{l-1}, g_l and g_r, g_{r+1} separately, and test whether some point of J is reachable from p_i and p_j, in constant time. (These two cells are the same if the middle part is empty, i.e. $l - 1 = r$.)

It may happen that g_l or g_r intersect J twice, and both pieces outside J are reachable in W_i. In this case, it is certain a valid biarc exists, and we need not proceed.

Lemma 2.3. *Let g be a gate that satisfies* (In) *and* (Out) *and intersects J twice, and suppose that both pieces outside J are reachable in W_i. Then there is a valid biarc between p_i and p_j.*

Proof. Let s_1, s_2 be two arcs in W_i, such that s_1 reaches one side of the gate g outside of J and s_2 the other. F denotes the segment on g between the two intersections of J (see Figure 2.12). If s_1 or s_2 reaches g as part of the backward visibility segment on g, then it must reach the same outer piece of g as part of the forward visibility segment. We can thus assume that s_1 and s_2 extend from p_i until they hit g for the first time. The region bounded by s_1, s_2 and the segment

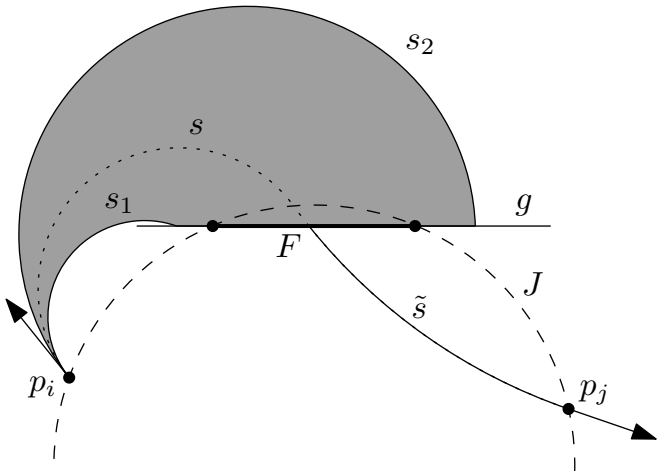


Figure 2.12: A joint circle J intersecting a gate g twice. The shaded region is contained in R.

g is contained within R, and therefore the part of J between the intersections with g is completely in W_i. Let us sweep the circular arc s from s_1 to s_2. Each of these arcs is a valid arc and it can be extended to a biarc ending in p_j (not necessarily valid). These biarcs sweep over the segment F. By condition (In), we know that at least one of the complementary arcs $\tilde{s}$ is a valid arc, at least to the point where it hits F. Since the region between F and J lies within R, the whole biarc is in R.

We show that this biarc is also valid biarc, since it does not intersect g_i and g_j except at p_i and p_j: By construction, the first arc s (up to the joint on J) does not intersect g_i twice, since its endpoint is in W_i, and it does not intersect g_j at all, since it terminates before crossing g. Similarly, the second arc $\tilde{s}$ does not intersect g_j twice, its endpoint being in separated from g_j by g. It could possibly intersect g_i only in the circular segment between F and J, but since this region is part of W_i, this is impossible.

□

So, the final case that we have to deal with is the following. We have a sequence of gates $g_l, g_{l+1}, \ldots, g_r$, that satisfy (In) and (Out). We therefore know that g_l and g_r (as well as all intermediate gates) intersect J in points Q_l and Q_r (see Figure 2.13). It may happen that g_l or g_r intersects J twice, but then only one of the outer parts is intersected by W_i. (Otherwise we are done, by the previous lemma.) We denote by g_l^{out} and g_r^{out} that outer segment of g_l and g_r that is intersected by W_i. The intersection Q_l and Q_r is chosen (in case there are two intersections) as the one that is incident to g_l^{out} and g_r^{out}, respectively.

We know that there is a valid arc from p_i to g_r^{out}. If the arc reaches g_r^{out} as part of the backward visiblity, it must have passed through g_r^{out} as part of the forward visibility region. Thus we denote the first intersection of the arc with g_r^{out} by S_r^{out}. Similary, there is an arc from p_j that reaches g_l inside J for the

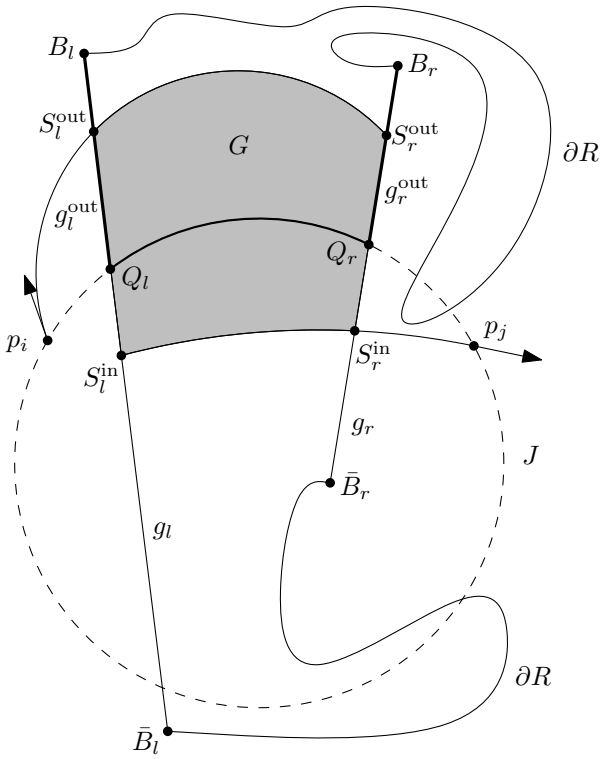


Figure 2.13: The region $\hat{R}$ between g_l and g_r

first time in some point S_l^{in}, after passing through g_r in the point S_r^{in} inside J.

Now, the two segments $S_l^{\text{in}}S_l^{\text{out}}$ and $S_r^{\text{in}}S_r^{\text{out}}$ and the two circular arcs $S_l^{\text{out}}S_r^{\text{out}}$ and $S_l^{\text{in}}S_r^{\text{in}}$ are contained within R. It follows that the four-sided region G enclosed by these curves (shaded in Figure 2.13) contains no part of the boundary of R, and in particular, the arc Q_lQ_r of the joint circle that lies in this region is not intersected by the boundary of R.

Lemma 2.4. *In the situation described above, a joint in the region R between g_l and g_r can only lie on the arc Q_lQ_r.*

This lemma seems obvious at first sight, but it is conceivable that this region contains parts of J besides the arc Q_lQ_r, as in the example of Figure 2.13.

Proof. Let $\hat{R}$ denote the region R between gates g_l and g_r. We denote by B_l and B_r the endpoints of g_l^{out} and g_r^{out}, and by $\bar{B}_l$ and $\bar{B}_r$ the opposite endpoints of g_l and g_r. Since the region G lies inside $\hat{R}$, it follows that the boundary of $\hat{R}$ must connect B_l with B_r and $\bar{B}_l$ with $\bar{B}_r$. (The opposite connection, B_l with $\bar{B}_r$ and $\bar{B}_l$ with B_r, would lead to a crossing.)

A valid arc starting from p_i enters $\hat{R}$ through g_l^{out}. This arc is then in the region $\hat{R}^{\text{out}}$ that is bounded by g_l^{out}, g_r^{out}, the arc Q_lQ_r, and the boundary of $\hat{R}$ between B_l and B_r. The arc may leave this region through g_r^{out}, but then it has to reenter through g_r^{out} in order to become a valid arc ending in $\hat{R}$. If the arc hits Q_lQ_r it terminates there. The arc must therefore meet J in the region $\hat{R}^{\text{out}}$.

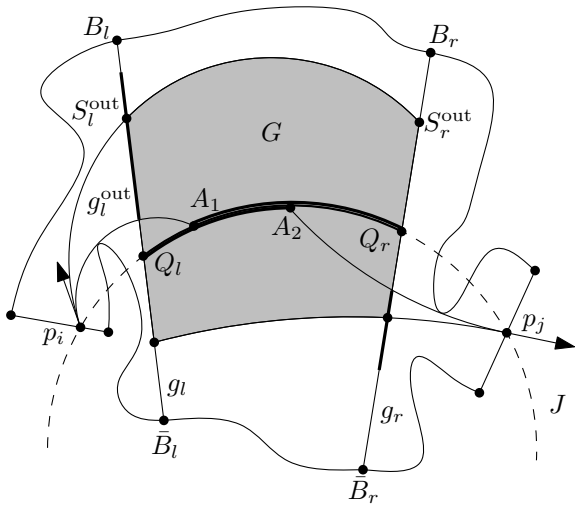


Figure 2.14: Determining the valid joint points

We can apply a similar argument for the backward arc from p_j. This arc is caught in the complementary region $\hat{R}^{\text{in}}$ that is bounded by the arc Q_lQ_r, the boundary of $\hat{R}$ between $\bar{B}_l$ and $\bar{B}_r$, and part of the segments g_l and g_r. Since the regions $\hat{R}^{\text{out}}$ and $\hat{R}^{\text{in}}$ intersect only in the arc Q_lQ_r, the joint can only lie on this arc. □ □

Now we can easily determine the points on the arc Q_lQ_r that are joints of valid biarcs.

We have established that the region G does not contain any obstacles. Now consider the point on g_l^{out} closest to Q_l that lies in W_i, and extend the arc from p_i through this point until it hits J in some point A_1 (see Figure 2.14). It may happen that $A_1 = Q_l$ if this point is in W_i.

If A_1 lies beyond Q_r, we conclude that no arc can reach Q_lQ_r, because such an arc would have to intersect g_l^{out} closer to Q_l. Otherwise, by stretching the arc and sweeping out till the arc $p_iS_l^{\text{out}}S_r^{\text{out}}$, we see that the complete interval between A_1 and Q_r is reachable from p_i by an arc that stays inside R. (It also follows that A_1 cannot lie before Q_l: by the above argument, Q_l would then be in W_i, and $A_1 = Q_l$ would have been chosen instead.)

Similarly, we can look for the point on g_r that lies in $\tilde{W}_j$, inside J, and is closest to Q_r, and we extend the arc from p_j to a point A_2 on J. We conclude that the whole sub-arc Q_lA_2 is reachable from p_j, or that no point on the arc Q_lQ_r is reachable.

By intersecting the arcs A_1Q_r and Q_lA_2, we eventually find the arc A_1A_2 of possible joints, or we find that no joints are possible. The joints on the arc A_1A_2 correspond to biarcs that lie in R. To get *valid* biarcs, we have to ensure that they do not intersect g_i and g_j other than in their endpoints. It is straightforward to reduce the interval A_1A_2 in order to exclude the biarcs violating this condition, in constant time.

We summarize what we have achieved.

Lemma 2.5. *After the intersections of all visibility regions W_i and $\tilde{W}_i$ with all gates g_k have been computed (Lemma 2.1), the existence of a valid biarc between two given endpoints p_i and p_j can be tested in $O(\log n)$ time.*

Proof. We do binary search on the gates between g_i and g_j to find the locations of g_l and g_r. This is the part of the procedure that takes $O(\log n)$ time. If $r < l-1$ there are no valid biarcs between the gates. Otherwise we test the cell between g_{l-1} and g_l and the cell between g_r and g_{r+1} to see if any part of J within these cells is a joint for a valid biarc. (The two cells could be the same cell.) Because the cells have constant complexity these tests can be done in constant time. We check if g_l or g_r intersects J twice and W_i intersects the gate on both sides of J. If so there is a valid biarc. If not, we compute A_1 and A_2 as described above and see if the arc A_1A_2 is non-empty. If not, there is no valid biarc. If so, we reduce it if necessary to eliminate biarcs that intersect g_i or g_j in points other than their endpoints. If any points remain in the interval we report that a valid biarc exists. All work after finding g_l and g_r requires $O(1)$ time. □

With the help of this test, we can now define the directed graph of reachable arcs, and the shortest path will give us the approximation with the fewest biarcs:

Theorem 2.1. *Given an open polygonal curve $P = (p_1, \ldots, p_n)$, a polygonal tolerance boundary of size $O(n)$, a gate and a tangent direction for each p_i, we can approximate P by a minimum number of valid biarcs in $O(n^2 \log n)$ time and $O(n^2)$ space.*

Proof. For each point p_i, determine which part of every other gate is reachable by computing W_i and $\tilde{W}_i$ in $O(n^2)$ time and space. For each point pair p_i, p_j we check whether a valid biarc exists, by Lemma 2.5. This requires $O(\log n)$ time, for a total run time of $O(n^2 \log n)$. We use these tests to set up a directed acyclic graph and compute the shortest path, in $O(n^2)$ time. □

As for arcs, there are instances where this bound is overly pessimistic. The visibility regions W_i and $\tilde{W}_j$ will often not extend beyond a few gates, and the calculation can be shortcut.

The biarc as well as the arc algorithm can be easily adapted to solve the problem of finding the minimum number of arc/biarcs approximating a closed polygonal curve. The main bottleneck of both solutions would be finding the shortest cycle in the graph of all valid arcs/biarcs. Using the results for computing the all pairs shortest paths by Zwick [80] would lead to a total $O(n^{2.575} \log n)$ runtime for both algorithms.

2.5 The Tolerance Boundary

The "approximation error" ε enters our problem only through the tolerance region R. The definition of a useful tolerance boundary for a given curve is a modeling question that depends very much on the application. For some applications, like cutting, it makes sense to use *asymmetric*, one-sided tolerance boundaries. In these settings it can be also more useful to not use the points of the original polygonal curve as start and endpoints for the arcs/biarcs. Both of our algorithms can deal with any sequence of ordered points as input points for the algorithm, if they are inside the tolerance region and in consecutive order form the beginning to the end of the tolerance region, e.g., so called "a-nodes" defined in terms of the tolerance region. They take the role of the points p_i. For a detailed introduction into computing tolerance regions and a-nodes, see for example the results of Held and Heimlich [46]. As long as the boundary meets the requirements specified in the beginning of this chapter, our algorithm can deal with it. Note that the width of the tolerance boundary may change within R. Therefore depending on the tolerance boundary our algorithm can answer the classical question for ε approximation (our solution has absolute guarantees for the minimum number of biarcs that are at most ε away from the original curve), as well as for approximations with changing precision requirements. More sensitive parts of the polygonal curve can be approximated with smaller ε than less important ones. Allowing variations of the width of the approximation boundary has no impact on the theoretical complexity of the problem. This ability to vary the width of the approximation boundary makes our algorithm useful for smoothing paths in robotics motion planning. The corridor used for the robotics is usually defined by the obstacles which have to be avoided. This leads naturally to corridors with non-constant width.

Part II

Approximation of polytopes in 3-dimensional space

Introduction

In this second part of the thesis we move on from 2-dimensional to 3-dimensional space. We deal with recovering structural information for a 3-dimensional object that is represented by a sample point cloud. The motivation for studying these problems is based on open problems in object simplification and surface reconstruction, two fundamental challenges in several areas of computer science, e.g., computer graphics and geometric modeling [1, 3, 7, 8, 14, 27, 29, 40, 45, 48, 54, 57, 65]. A lot of research has been done in the field of approximation of three-dimensional point sets with polytopes with surfaces of higher order [35, 55], e.g. Bézier surfaces [49, 71]. Recent advances in graphics hardware allow the rendering of algebraic surfaces by their polynomial representation without computing approximating triangles [56].

We initiate the study of this problem by considering convex surfaces only, for simplicity, before moving on to non-convex surfaces. A first natural step to higher-order approximation of convex polytopes is the approximation with spheres or spherical patches. Since polyhedral facets can be seen as spherical patches with infinite radius, spherical patch approximation is a generalization of polytope approximation and a natural followup to the biarc algorithm. Most approaches to modeling with balls or spheres are closely related to the *union of balls* [37, 35] (as is our approach to non-convex polytopes). Our goal is to model convex or nearly convex parts of a surface by a convex surface formed by spherical patches. This approach is closely related to the *intersection of balls*. In particular the resulting surface can be used for separating convex shapes, i.e., for constructing bounding volumes.

The polytopes used in this theses are solids bounded by a surface. The surface is a 2-manifold and each face of the surface is a simple planar polygon.

We require that the surface is triangulated, closed and bounded. We make the following assumptions about the intersection of components and the local topology (similar to [64]):

- For each pair of faces, we require that either
  - they are disjoint, or
  - they have a single vertex in common, or
  - they have two vertices, and the edge joining them, in common.
- The neighborhood of each vertex is homeomorphic to a disk. Every vertex is the apex of a number of triangles. The link of a vertex v is the collection of edges opposite v in all triangles incident to v. We require that the link of every vertex be a simple, closed polygonal path.

A *convex polytope* is a polytope, which lies entirely on one side of any plane defined by its faces. Each face is a simple convex polygon.

In Chapter 3 we can show that deciding the existence of an approximation of a convex polytope with a given upper error bound ε and not more than a given number of spherical patches is NP-hard. The complexity of the approximation problem is closely related to open problems in polytope approximation, in particular to the complexity of polytope approximation with the minimum number of facets. In future we hope to use our new methods from the NP-hardness proof of the more generalized problem to solve the complexity question of the minimum facet polytope approximation.

In Chapter 4 we present a new technique for constructing curved surfaces based on inscribed polytopes resulting in a convex surface consisting of spherical patches. The inscribed polytope approach makes our construction suitable for various incremental algorithms.

These results on convex polytope approximation were presented at the 23nd European Workshop on Computational Geometry (EWCG) in Graz, in March 2007 [19].

To tackle the approximation problem for non-convex polytopes we pick up the idea of an incremental approximation algorithm. This induces the problem of finding a simple and topologically correct start polytope, the 'seed polytope', for non-convex polytopes. Our work is strongly related to the results of Amenta and Bern [7] and is based on several of their results.

For a surface $\mathcal{F}$ in 3-space that is represented by a set S of sample points, we construct a coarse approximating polytope P that uses a subset of S as its vertices and preserves the topology of $\mathcal{F}$. In contrast to surface reconstruction we do not use all the sample points, but we try to use as few points as possible. Such a polytope P can then be used as a seed polytope for starting an incremental refinement procedure to generate better and better approximations of $\mathcal{F}$ based on interpolating subdivision surfaces or, e.g., Bézier patches.

Our algorithm starts from sample S (an r-sample, which is defined in Chapter 5) of $\mathcal{F}$. Based on S, a set of surface covering balls with maximal radii is calculated such that the topology is retained. From the weighted α-shape of a proper subset of these highly overlapping surface balls we get the desired polytope. As there is a rather large range for the possible radii for the surface balls, the method can be used to construct triangular surfaces from point clouds in a scalable manner. The seed polytope algorithm and the surface reconstruction are joint work with Oswin Aichholzer, Franz Aurenhammer, Bernhard Kornberger, Simon Plantiga, Gert Vegter and Günter Rote, we describe the algorithms in Chapter 5.

These results on non-convex polytope approximation have been accepted for presentation at the Eurographics Symposium on Geometry Processing (SGP) 2009 (July) in Berlin [6]. The algorithm has been implemented by Bernhard Kornberger.

Chapter 3

Approximation by Spherical Patches: An NP-hardness proof

The optimization problem we are considering is the *Approximation by Spherical Patches* (ASP): the approximation of a convex point set with a given number g of spherical patches resulting in a convex surface with all points within some specified tolerance to the surface. We show that deciding if a surface can be approximated by g patches is NP-hard. In our reduction from grid 3SAT, we use consecutive patches to encode the Boolean information by alternating flat and bulbous patches. The same decision problem for facets of an approximating *polytope* is assumed to be NP-hard due to its close relation to the minimum facet separator problem. It is known that finding a minimum facet separator for two convex polytopes is NP-hard [25, 24, 26, 62]. A polytope can be approximated by separating the inner and outer boundary of the "fattened" polytope surface. This special case of the minimum separator problem remains an open problem [2]. Since a convex polytope can be seen as the intersection of spherical patches with infinite radius our NP-hardness proof resolves a generalized version of the problem.

Problem Statement. We are given a set P of n points in convex position. We want to find a convex approximating surface S that consists of *spherical patches*. A spherical patch is part of the boundary of a ball. There are two quality criteria that we want to optimize: (a) the approximation error, which is defined as the maximum distance from a point of P to S; and (b) the number of patches.

Our approach is closely related to the intersection of balls. In our specific construction, we use only spheres with the following constraints: some points of P lie on each sphere, the spheres are large enough to contain all other points of P inside. Thus, we can construct the approximating surface simply as the boundary of the intersection of all approximating balls and halfspaces (balls with infinite radius) that contribute in the construction.

Outline Starting from a planar 3SAT instance, we construct a point set defining an instance of the ASP with zero tolerance such that - in the satisfiable case - a minimal solution of the approximation problem corresponds to a truth assignment in the grid-3-SAT problem. We split the grid in four types of grid cells: fill cells, variable cells, wire cells and clause cells. Further we associate point triples to the first three types of cells, and single points to the third type. This point set is then lifted to a paraboloid and extended with additional points (to be specified later).

The points are constructed such that they can be approximated with one spherical patch for each grid cell, except for the clause cells. The points corresponding to clause cells will be covered by this set of patches, exactly if they correspond to a truth assignment to the grid 3SAT instance. Finding an approximation with a minimal number of patches therefore results in finding a truth assignment for grid 3SAT.

We describe the minimal solution in the satisfiable case and prove that more patches than g are needed in the non-satisfiable case for the ASP.

Our approximating surface can be seen as the surface of intersection of g spheres. Therefore the boundaries of the patches are defined by the intersection curves of neighboring patches.

3.1 Grid-3-Satisfiability

3-SAT statements consist of a Boolean conjunction of clauses, where each clause consists of a disjunction of three Boolean variables, each of which may be negated. Such a statement can be represented by a bipartite graph, where variables and clauses are represented by vertices. Each clause vertex is connected to its three variable vertices by an edge marked $+$ or $-$ depending on whether this variable occurs negated in that clause (see Figure 3.1). Planar 3-SAT consists of all 3-SAT problems represented by a planar graph. Finding a valid variable assignment is NP-hard. Furthermore, *Grid-3-Satisfiability (grid 3SAT)* is NP-hard [42, 74] where the variable-clause graph of a formula of length n in 3-conjunctive normal form can be embedded on a $c_1 n \times c_1 n$ grid with c_1 some constant.

3.2 Modifying the grid

We start with a grid of squares of side length 1. The first step of the reduction requires a refinement of the grid by factor $c_2 = 30$ (see Figure 3.2).Vertices correspond to facets and edges correspond to rectilinear paths on the grid. Further we disperse the grid cells by a small constant factor $\delta = 0.1$ which creates gap rectangles between the cells. Depending on the label of the edge in the variable clause graph, we change the number of facets in the path on the grid corresponding to the edge. A negatively labeled edge is represented by a path with an odd

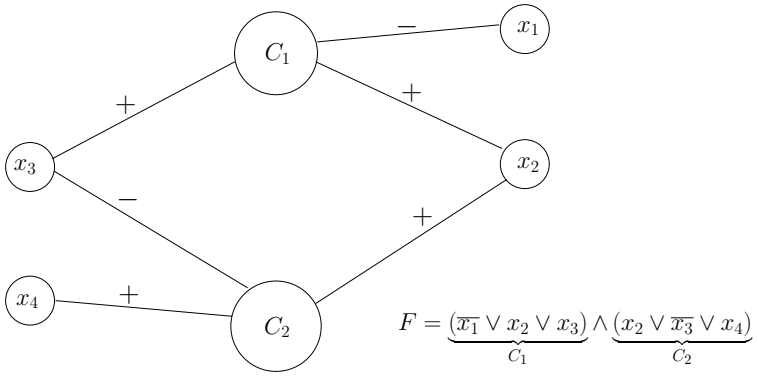


Figure 3.1: Planar embedding of a 3-SAT problem: The variable-clause graph of the formula F.

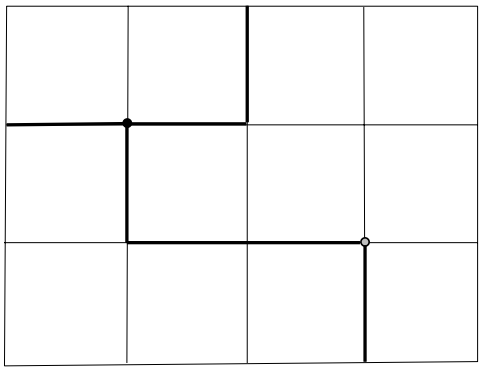
number of cells and a positively labeled edge corresponds to a path with an even number of cells. To achieve this correspondence we need sufficiently many cells on a straight path. The inclusion of an additional cell is done by reducing the gap size δ to δ' in a straight segment of the path. The size of δ' depends on the number of cells in the considered path segment, $0.064 < \delta' < 0.1$. We compute δ' in such a way that we gain a gap of size $1 + \delta'$, so that we can fit in another cell of unit size. We do the same procedure with the grid cells which are adjacent to the changed path segment. The grid has a size $30c_1n \times 30c_1n$ and consists of four kind of cells: variable cells, wire cells (cells of the paths), clause cells and fill cells. Next we delete all grid cells corresponding to clauses (the clauses will be represented later by a single point). We also drop the lower right vertex of each grid cell. For each grid cell i we have therefore a triple P_i.

Lifting to a paraboloid The next step is a lifting of the point triples of the grid cells onto a very flat paraboloid. The paraboloid depends on the size of the grid. We pick a paraboloid of the form $z = -\lambda \cdot (x^2 + y^2)$. (We use the mirrored paraboloid for easier illustration.) Let D_i be the circumcircle of the lifted point triple P_i. The parameter λ has to be chosen in such a way that for two neighboring point triples P_i, P_j the disks of D_i and D_j intersect (their circumcircles do not necessarily intersect), see Figure 3.3. This guarantees the existence of valid spherical patches.

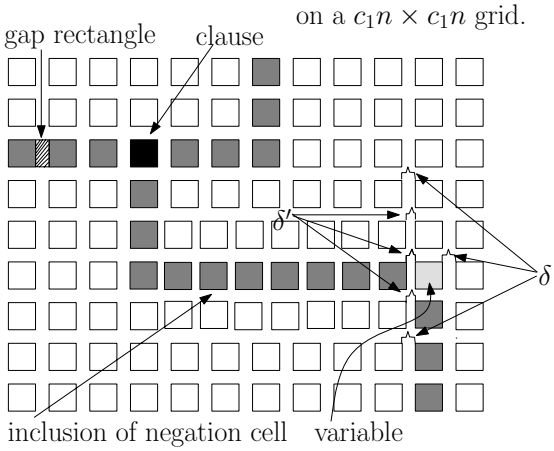
For a lifting of a δ dispersed $cn \times cn$ grid, with $c = c_1 \cdot c_2 = 30c_1$, this leads to a bound on λ:

$$\lambda < \frac{\sqrt{\left(1 - 1/\sqrt{2}\right)^2 - \delta^2}}{\delta^2 + 2\delta(c + \delta)n}.$$

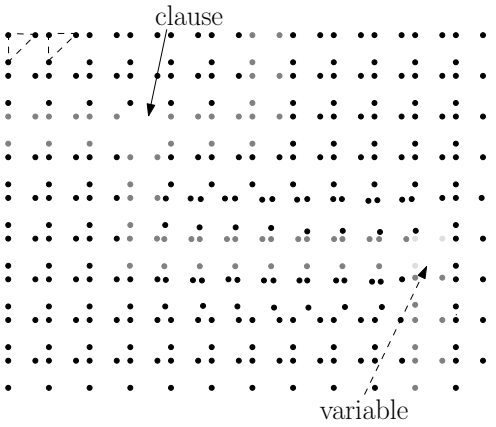
In order to satisfy this inequality, δ has to be chosen less than $1 - 1/\sqrt{2}$. This gives us a point set in convex position in 3D (see Figure 3.4). For our explicit construction we choose $\delta := 1/10$ and $\lambda := 1/(10m)$ with $m = c(1+\delta)n$ a bound on the width and length of the dispersed grid (see Theorem 3.1 below).



(a) Schematical figure of the embedding on a $c_1 n \times c_1 n$ grid.



(b) Schematical figure of a refined grid. For clarity, the grid is only refined by a factor of 3.



(c) The associated point set to (b).

Figure 3.2: Refinement of the grid.

3.3 Fill points

Looking at the lifted point set we place additional points: We place one point into each triangular face corresponding to fill cells. We refer to these four points as a set of fill-points.

Lemma 3.1. *Each set of fill points induces exactly one spherical patch and all sets cannot be covered with less than one patch per set.*

Proof. A convex surface containing the four points must contain the whole triangle. Hence the spherical approximation must contain a planar face that includes this triangle.

□

There are five more facets of the convex hull of the lifted point set to consider: the flat bottom facet with the four adjacent border facets. To ensure that they get covered with exactly one patch each, we place four additional, non co-circular points into each of them.

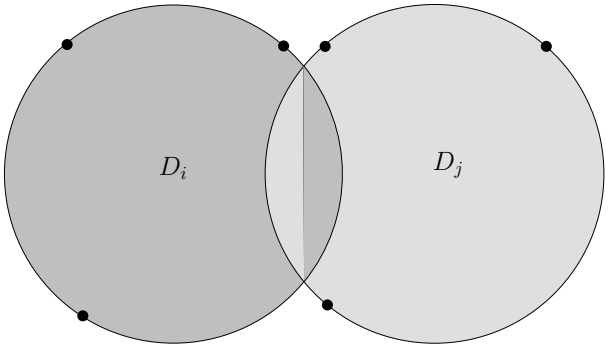


Figure 3.3: Intersection of the disks of D_i, D_j. The disk are induced by the lifted point triples P_i, P_j.

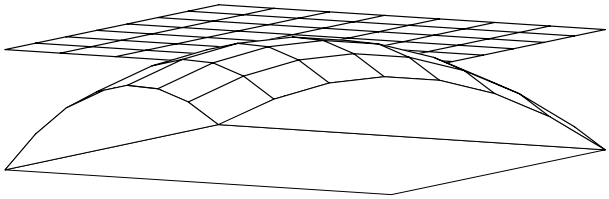


Figure 3.4: Lifting of the (undispersed) grid to a paraboloid.

3.4 Wire

The wire corresponds to edges in the variable-clause graph, which are represented by sets of lifted point triples. We build a wire out of consecutive spherical patches to propagate information from the variables to the clauses. We will prove that, if there is a solution with g patches, each lifted point triplet will be covered by a single patch. The main idea of the reduction is to place additional points between consecutive triplets in the wire, that narrow down the choice to only two spheres for each patch: a flat patch or a bulbous patch. We choose for a flat patch the sphere with center at infinity, the plane defined by the lifted point triplet P_i. The common radius of the bulbous spheres is $r := 10m$ (see Theorem 3.1 below).

We place a set of four points, P^{FF}, on the intersection line of consecutive flat patches, four points, P^{FFill}, on each intersection line of flat patches and their adjacent fill cell patches and one point, P^{BB}, on the intersection circle of bulbous patches, as described below. These additional points force the patches to alternate between flat and bulbous patches (see Figure 3.5).

To avoid exact computation of the intersection of two bulbous spheres we place the point P^{BB} approximately on the intersection. We deal with a specific $\varepsilon = 1/m^2$ that is calculated carefully, ensuring that all previous lemmas still hold (see Theorem 3.1 below). Regarding Lemma 3.1, note that ε is chosen in such a way, that no patch can contain two or more whole triangles.

The bulbous spheres are large enough to contain all other points inside. Thus, we can construct the approximating surface simply as the boundary of the in-

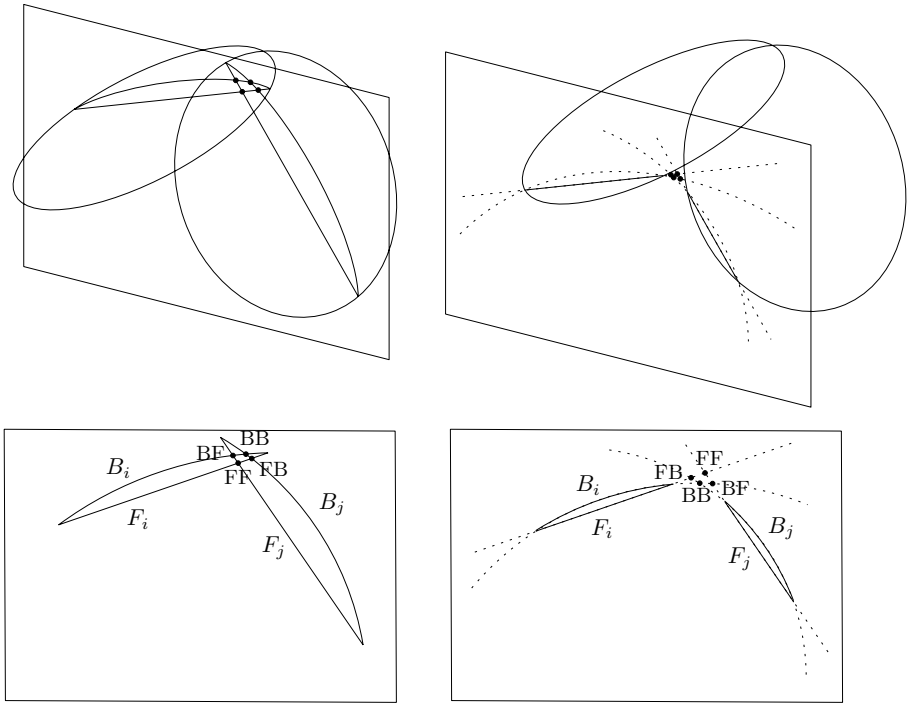


Figure 3.5: Two different cross-sections through two adjacent patches, which can be flat (F_i and F_j) or bulbous (B_i and B_j), respectively. In the central cross-section (left), an FF intersection point is covered by all pairs of patches except B_iB_j. In the off-center cross-section (right), a BB intersection point is covered by all pairs of patches except F_iF_j.

tersection of all approximating balls and halfspaces that contribute in the construction.

We need to guarantee that the additional points, P^{FF}, P^{FFill} and P^{BB}, on the flat and bulbous patches will lie on the approximating surface (see Lemma 3.2). The points P^{FF} lie on the intersection of the disks (the disks corresponding to the circumcircles of the point triples) and the point P^{BB} lies on the circular arc outside the intersection (see Figure 3.6). The point sets P^{FFill} are placed on the intersection lines of the flat wire patches and their neighboring fill cell patches (see Figure 3.7).

Construction of the point P^{BB}. If at least one of the two bulbous spheres is chosen the point must lie on the intersection surface of the construction. In the following we formulate conditions under which a point lies on the intersection surface. Then we pick such a point and prove that the conditions hold.

Since the radii of the bulbous spheres have been chosen in such a way that they only come close to the grid polytope at the face by which it is defined, it suffices to consider the local configuration. The *grid polytope* is the convex hull of all the lifted variable and wire grid vertices.Thus, for a point to lie on the

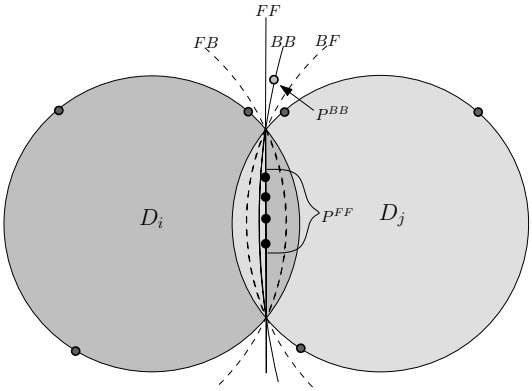


Figure 3.6: Top view of two neighboring wire cells: Placing of the points on the FF and BB patch intersection. The light gray point is on BB.

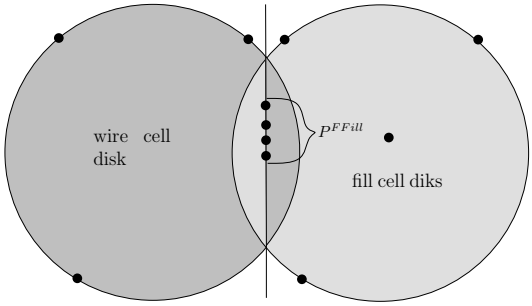


Figure 3.7: Top view on adjacent wire/fill cell:Placing of the points P^{FFill} on the intersection of flat patches and fill cell patches.

intersection surface the following conditions are sufficient:

(1) The point lies below both of the planes defined by the two triplets of points.

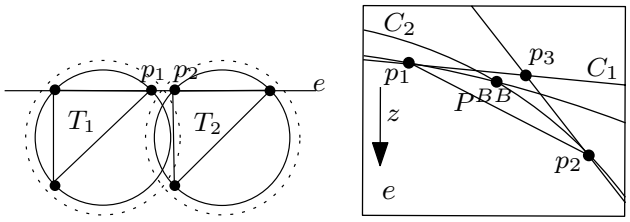
(2) The point lies above the face of the gap rectangle between the two faces defining the spheres.

(3) The projection of the point lies within (possibly on the boundary of) the gap rectangle between the two faces defining the spheres.

These conditions guarantee that the point chosen will lie on the convex hull of the point set and that all other points remain on the convex hull.

For two neighboring triplets of points there are two points p_1 and p_2, one of each triplet, neighboring in the grid (see Figure 3.8 (a)). Consider the vertical plane e through these two points. We place as the point P^{BB} the intersection point of this plane and the two spheres outside of the grid polytope.

The point P^{BB} fulfills the third condition by definition. To see that the other two conditions hold consider Figure 3.8 (b). The two triplets of points define two planes e_{T_1} and e_{T_2}. These two planes and the plane e intersect in a point p_3. The triangle defined by p_1, p_2, and p_3 lies in the plane e. The intersection



(a) Vertical View

(b) Configuration in the plane e

Figure 3.8: Construction of a valid intersection point of the "bulbous" spheres.

point of the two spheres and plane e fulfills conditions (2) and (3) if it lies in this triangle.

Intersecting each of the two spheres with e yields two circles C_1 and C_2. Since p_1 lies on the sphere, C_1 passes through p_1. The circle C_1 cannot cross the edge (p_1, p_3) and by the construction of the spheres does not cross (p_1, p_2). Thus it intersects the edge (p_2, p_3). Analogously C_2 passes through p_2 and intersects (p_1, p_3). Therefore, the circles intersect in the triangle.

3.5 The variable and clause gadgets

Variable. A variable is a point triplet which is handled as a wire point set. Choosing the flat patch corresponds to a *false* assignment and the bulbous patch to a *true* assignment. A flat patch results in a covering of all points P^{FF} around the variable point triplet, therefore all consecutive wire patches will propagate the same information – all wires starting from this variable will start with a bulbous patch. Picking a bulbous patch for a variable point triplet is symmetric.

Clause. Before the lifting, a clause corresponds to a grid cell in the plane which is connected to three wires (from three variables). We deleted the vertices of the clause grid cells, they do not contriubte to the construction. We place a single point in the free space between the three wire point triplets, at the intersection of the bulbous patches of these point sets (Note that because of the bigger free space the intersection of the bulbous patches lies below the intersection of flat patches in contrast to the situation between wire patches). Like for the point P^{BB}, to avoid exact computation of the intersection of now three bulbous spheres we place the point approximately on the intersection, with $\varepsilon = 1/m^2$.

3.6 NP-hardness proof

To prove that an approximation of the constructed point set with a minimum number of patches corresponds to a truth assignment in the grid-3-SAT problem

we need to show three things:

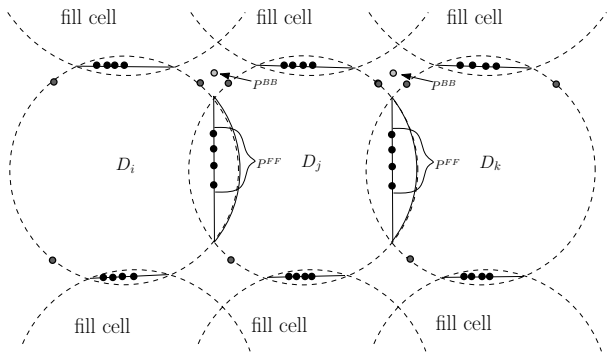
- The surface constructed only out of fill cell patches and alternating wire patches covers the entire constructed point set.
- Only alternating wire patches can lead to a minimum number of patches for the surface overall.
- There exists a satisfying assignment for the grid 3SAT instance if and only if the lifted point set with all additional points can be approximated with g spherical patches. For a $30cn \times 30cn$ grid, with k clauses and t cells included for negation, $g = 30cn \times 30cn + t - k + 5$

Lemma 3.2. *If we choose alternating flat and bulbous spherical patches in every wire, all point sets P^{FF}, P^{FFill} and points P^{BB} are on the approximating surface.*

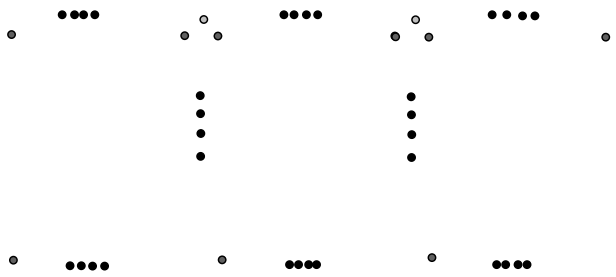
Proof. The point sets are only placed between consecutive point triplets in the wire (variable patches are handled as wire patches) and between adjacent wire/fill cells. The intersection of the wire patches can be of two kinds: flat/bulbous or bulbous/flat. Consider points P^{FF} placed on the intersection line of two adjacent flat patches F_i and F_j. If we shrink the radius of F_j, the patch becomes more bulbous. The intersection of the two patches moves continuously toward F_j. If we shrink instead the radius of F_i, this patch grows more bulbous and the intersection moves toward F_i. So the intersection of the two patches moves away from the points P^{FF} toward the bulbous patch. Therefore the points P^{FF} stay on the part of the flat patch that contributes to the surface (see Figure 3.5). Symmetric arguments hold for P^{BB} and for the point sets P^{FFill}. The fill cells always have a flat patch, therefore the intersection of the wire patch with the fill patch can only be either flat/flat, in this the case the points are on the surface by construction, or flat(the fill patch)/bulbous(the wire patch). For the second case the intersection curve of the two patches moves toward the bulbous patch and the points P^{FFill} are on the part of the fill patch that contributes to the surface. □

Lemma 3.3. *The minimum number of patches for a wire with l point triplets is l. This minimum can be achieved only if each wire triplet is covered by a separate patch and adjacent wire patches have different types, either bulbous or flat.*

Proof. Point triplets neighboring a wire triplet correspond either to fill grid cells or to the same wire, because of the embedding of the 3SAT instance on the dispersed grid. Each wire triplet has exactly two neighboring fill cells. According to Lemma 3.1, we need exactly one patch for every set of fill points and because of the convexity of the point set, these patches cannot cover any point of the wire point triplets. So a wire is a band of points, the point triplets plus the additional points we placed, see Figure 3.9.



(a) Three neighboring wire triplets with fill cells.



(b) The set of points in this wire part.

Figure 3.9: The points in the wire induced by three wire grid cells. All of these points have to be on the approximating surface.

All points in a band are in convex position and we require that all of these points lie on the surface. As shown in Lemma 3.2 we constructed the additional point sets in such a way, that alternating flat and bulbous spherical patches in the wire fulfil this request. To show that any other choice of patches leads to more patches in total we make a case distinction. The essence of the minimum patch construction lies in the alternation of the patches. Therefore we need to look at the number of points and the length of the band we can cover with two alternating patches and show that with no other patch construction we can cover the same amount of points or band size with two or less patches.

Counting the number of points covered with two alternating patches: By construction a bulbous patch covers the defining triplet P_i and additionally the points P^{BB}, so five points in total. Furthermore, as shown in Lemma 3.2, the triplet adjacent two sets P^{FFill} are on the part of their fill patches that contributes to the surface. A flat patch covers the defining triplet P_j and additionally the two set of points P^{FF}, the two sets P^{FFill} are on the intersection line of the flat patch with the fill patch and contribute therefore to the surface. So the alternating patches together cover all points in the band which are adjacent to

two neighboring wire triplets, a total of 16 points. Additionally all adjacent sets of P^{FFill} contribute to the surface as required, so the complete wire band spanning these two triplets is covered with two patches. We denote this covered wire band by η.

Covering the points in η with two bulbous or two flat patches: A bulbous patch defined by P_i cannot cover any additional point out of P^{FF} because they are by construction inside the disk D_i (see Figure 3.6) and therefore inside the bulbous patch. Two bulbous patches cover only their defining wire triplets plus three points P^{BB} which is not only just a total of 9 points, but also the set of points P^{FF} between the triplets P_i, P_j does not contribute to the surface and at least one more patch is needed to correct this. So we need a total of 3 patches. As the points P^{BB} are not on the same plane as the point triplets P_i, P_j, any two flat patches covering these triplets would not cover P^{BB}. Although two neighboring flat patches would cover a total of 18 points, we have a point P^{BB} in the band between the patches that is not covered and therefore requires an additional patch. So we need a total of three patches two cover the same length of wire band we can cover with just two alternating patches.

Covering the points in η with a set of patches where at least one patch is none of our constructed patches: Lets assume we could cover the same length of wire band with two or less patches, at least one of which is not defined by a wire triplet (P_i is left of P_j as in Figure 3.9). The points P^{FF} between the triplets are collinear, so any construction covering these points with a bulbous patch could always just cover a maximum of two points out of the set of four, and only one point of the triplet P_j. The point triplet P_j and the adjacent points P^{FF} lie in the same plane by construction, therefore any combination of three non collinear points define a circle, any sphere through this circle covers only the points on the circle but no other points on the circle plane. Therefore a patch G could cover a maximum of 6 points if it covers also the points P^{BB}, to cover the same length of wire band we still need to cover two points of the P_j triplet plus at least a part of the P_i triplet. The only patch that can cover the two remaining points of the set P^{FF} and the triplet P_i is the plane through the wire triplet, therefore we need at least an additional patch to cover the remaining points of P_j. Constructing a bulbous patch to cover the remaining points of P_i, P_j instead of the flat patch leads always to a patch which can cover at most 4 of the remaining 7 points, which again will require at least one more patch, so a total of 3 patches to cover the point sets and the desired length of the wire band.

Constructing the patch G by taking no point out of a wire triplet, e.g. P_i, we can take a maximum of two points of the two P_i adjacent sets P^{FF} and the two points P^{BB}. The points of the sets P^{FF} and P^{FFill} are placed by construction inside the disk D_i (see Figure 3.9). Therefore we cannot cover the points of P_i

and the at least 4 uncovered points of P^{FF} with one flat or bulbous patch in such a way that all points contribute to the approximating surface. So it is easy to see that we need at least three patches to cover the wire band of P_i, P_j.

Any flat patch that is suppose not to cover a wire triplet, can cover only two points of the 11 coplanar points (the triplet plus the two sets P^{FF}), and at most one point of either an adjacent wire triplet or a point P^{BB} . It can be easily seen that these planar patches always lead to more patches in total as they leave to many points uncovered.

□

Theorem 3.1. *There exists a satisfying assignment for the grid 3SAT instance if and only if the lifted point set with all additional points can be approximated with g spherical patches. For a $30c_1n \times 30c_1n$ grid, with k clauses and t cells included for negation, $g = 30c_1n \times 30c_1n + t - k + 5$*

Proof. According to Lemma 3.1 we need exactly one patch for every set of fill points. By Lemma 3.3 each grid cell which contributes to the wire is approximated by exactly one spherical patch. Point sets representing variables count as wire sets, therefore each variable is covered with exactly one patch.

The point representing the clause is covered by the wire patches if and only if at least one literal of the clause is true. If no literal is true the clause needs an additional spherical patch. The last patch of the wire before the clause point represents the assignment of the literal. If a literal in the clause is true the last wire point triplet has a bulbous patch. A bulbous patch covers the clause point because the point is on the intersection of all three bulbous patches and therefore on the surface of each. No additional patch is needed for the clause. If no literal is true, the clause point is not on the surface of any of the last three wire patches. In this case at least one more patch is needed to cover the clause, corresponding to a false assignment.

Looking at the dispersed $30c_1n \times 30c_1n$ grid this gives us a total of $g = 30c_1n \times 30c_1n + t - k + 5$ with k clauses and t included cells for negation and 5 cells for the upper bounding facets of the convex polytope of the lifted point set. Only if all clauses have a truth assignment, all points representing a clause are already covered by the spherical patches of the wire. Therefore if the 3SAT instance has a truth assignment, we need exactly g patches to approximate the constructed point set. By exact computation we get the following bounds: For a SAT instance on a $30c_1n \times 30c_1n$ grid let P be the set of points in convex position constructed as above with $\delta := 1/10$, $\lambda := 1/(10m)$, $m := \sqrt{2}(1+\delta)30c_1n$, and the common radius of the bulbous spheres $r := 10m$. Let P' equal P with the exception that the points on the bulbous-bulbous sphere intersections might be displaced by $\epsilon := 1/m^2$. For sufficiently large m the following holds: If the SAT instance is feasible, then there is a surface with g patches such that all points have distance at most ϵ to a patch. If the SAT instance is infeasible, then for every surface with at most g patches there is at least one point which has distance

more than 100ϵ from all patches. For the values for δ, λ, m, and r we compute a bound of $1/(1000m) - 1/m^2 O(1)$ (by exact computation) on the distance between any point p in P and any valid sphere not corresponding to p. Since the points in P' are perturbed by at most $\epsilon = 1/m^2$ the bound still holds. □

Chapter 4

Convex Approximation by Spherical Patches

Since finding an optimal solution to the spherical approximation is NP-hard, it makes sense to try heuristic approaches for finding good solutions. In this chapter we present a construction of curved surfaces based on inscribed polytopes, resulting in a convex surface consisting of spherical patches. The inscribed polytope approach makes our construction suitable for various other incremental algorithms, for example incremental construction of an approximating subdivision surface. We start with an inscribed polytope consisting of a single tetrahedron, and incrementally extend this polytope until the corresponding surface is a valid approximation. We do not have a guarantee for the quality of the solution in terms of the number of patches required. Nevertheless, the construction involves some non-trivial geometric arguments, which make this algorithm interesting.

The approximating surface is generated by a convex triangulation, in particular the convex hull of a subset of the input points. The triangles of this hull are called *supertriangles*. Our goal is to 'inflate' this polytope by replacing its faces with curved, spherical patches that go through the vertices of each supertriangle. These spherical patches are the part of the caps contributing to the surface. Once we have fixed the polytope, the only thing that remains to decide is the radius of each sphere.

4.1 Constructing a curved surface

In order to produce a valid surface, we require that spherical caps fulfill two main conditions, explained in more detail later. Let a be a spherical cap that goes through the vertices of a supertriangle t_i.

1. Intersection condition: The intersection of the circumcircle of an adjacent supertriangle t_j with the outer halfspace of supertriangle t_i, has to lie outside the spherical cap a (see Figure 4.1), i.e., the spherical cap a should

pass below the circular arcs of its three neighboring triangles. This ensures that neighboring patches intersect properly, resulting in a closed, convex surface.

2. Pyramid condition: The supporting planes of the three adjacent supertriangles intersect in a pyramid above supertriangle t_i. We require that the cap does not intersect the edges of this pyramid. As we will see later, this ensures that each supertriangle corresponds to exactly one patch of the surface (Figure 4.2). This condition prevents quadratic complexity of the resulting surface.

First we construct the spherical caps. The supporting plane of each supertriangle t_i splits space into an inner halfspace H_i^+ containing the convex hull, and an outer halfspace H_i^- (see Figure 4.3 a). For each supertriangle we construct a spherical cap by first taking a sphere S_i through its vertices with its center in the inner halfspace. Then we take the intersection of this sphere with the outer halfspace.

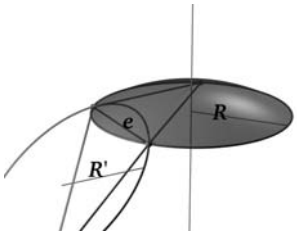


Figure 4.1: Supertriangle with spherical cap.

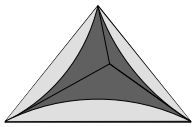


Figure 4.2: The dark gray area is the part of the patch inside the pyramid. The pyramid is defined by the intersection of the halfspaces of the three neighboring supertriangles.

4.2 Determining the radius

The center of the spherical cap has to lie on a halfline perpendicular to the supertriangle. For each neighboring supertriangle, the circumcircle and dihedral angle give a lower bound on the radius of the spherical cap, to ensure that the cap is flat enough to pass below that circumcircle (the intersection condition). Taking the maximum over the three adjacent supertriangles results in a single lower bound r_i^{min} for the cap radius. For technical reasons, we considere for each sphere S_i the union of the ball B_{S_i} with its inner halfspace H_i^+ defined by the

supertriangle. We define the approximating body as the intersection of these unions :

$$\bigcap_i (H_i^+ \cup B_{S_i})$$

This defines a bounded solid. We show that the halfspaces don't actually contribute to the boundary of this solid:

Lemma 4.1. *If neighboring spherical caps intersect properly, the intersection of the union of balls with their inner halfspace forms a convex solid.*

Proof. The three neighboring spherical caps of a supertriangle pass below its circumcircle, and so does the union of the corresponding balls/halfspaces. Therefore, the three arcs of this circumcircle don't contribute to the intersection of the union of balls/halfspaces (see Figure 4.3). All caps are convex and have proper intersections consisting of circular arcs. The circular arcs of the intersection are also convex, resulting in a convex surface. □

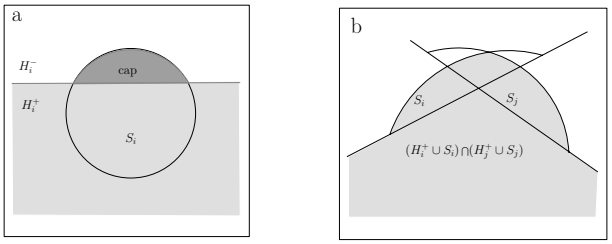


Figure 4.3: a) The union of a ball and the inner halfspace. The darker gray part is the spherical cap. b) The intersection of two unions of ball/halfspace.

4.3 Pyramid condition

Although this construction results in a valid spherical patch surface, we introduce the pyramid condition to deal with the following potential difficulty: As illustrated in Figure 4.4, it is possible that a cap contributes disconnected pieces to the surface. Here, the caps a and d are the flattest of the four caps. Cap b consists of a narrow, bulbous patch, that gets cut in two by patch d. In the worst case, this phenomenon can result in a number of patches quadratic in the number of facets of the inscribed polytope.

The intersection of the inner halfspaces of the three neighboring supertriangles of triangle i, defines a pyramid above this supertriangle. This pyramid lies inside the intersection of all inner halfspaces H_j^+, $j \neq i$, and therefore also inside $\cap_{j \neq i} H_j^+ \cup S_j$. As a result, the part of the spherical cap inside this pyramid contributes to the surface. We require that this cap passes below the three edges of this pyramid. Then, the part inside the pyramid is a single piece connected to

the three vertices of the supertriangle. As a result, the cap contributes a single connected patch to the surface. This pyramid condition can be expressed as another lower bound on the radius of the cap.

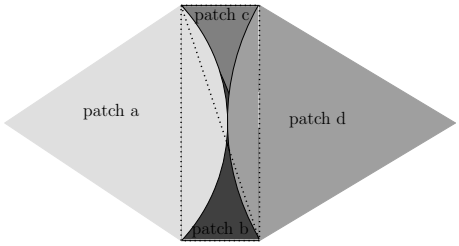


Figure 4.4: The two outer most patches (patch a and d) cut the dark gray patch b into two separated pieces.

4.4 Incremental construction

We now construct a curved convex surface from a subset S of the input points P. The convex hull of S generates a surface as long as the patch radii are chosen large enough, to ensure proper intersection (circumcircle property) and connectedness (pyramid condition).

For an incremental approach, we initialize S to the four extremal points of the point set P, in the directions of the normals of a regular tetrahedron. Respecting the lower bound on the radii, we try to choose cap radii such that the caps are closer than ϵ to the remaining input points. If this is not possible we add more input points to S.

It is not sufficient to test if every input point is ϵ close to some cap, since it would still be possible that they are not ϵ-close to the approximating surface of these caps, in particular if adjacent supertriangles have a large dihedral angle. We therefore require a slightly smaller distance δ for input points outside the caps. This δ is a fraction of ϵ depending on the dihedral angles of the polytope, increasing toward ϵ as the polytope gets refined and the dihedral angles decrease.

A supertriangle i is *valid* if there exists a corresponding spherical cap with radius larger than its lower bound r_i^{min}, such that all points inside the outer halfspace H_i^- of the supertriangle are closer than $\delta_i = \epsilon / cos(min\{\alpha_1, \alpha_2, \alpha_3\}/2)$ to this spherical cap, where $\alpha_1, \alpha_2, \alpha_3$ are the three dihedral angles incident to a supertriangle i.

Lemma 4.2. *If all supertriangles are valid, every input point is closer than ϵ to the approximating surface.*

Points inside the outer halfspace of a supertriangle do not always end up near the corresponding patch in the final surface, since adjacent patches cut

away parts of the patch. This means that we may test too many points for ϵ-closeness. For a sufficiently dense set of input points taken from a smooth convex surface however, we expect points close to a cap to be close to the approximating surface as well. Also, if points lie outside the circumcircle of a supertriangle, it might not be possible to find a valid cap at all. As an alternative to testing too many points, we could reduce the set of points we have to test for ϵ-closeness, by examining the neighboring supertriangles. Again, for smooth input this is not expected to happen.

For a sufficiently dense sample of a smooth surface we expect points ϵ-close to caps to be sufficiently close to the approximating surface as well, since we expect the approximating surface to have small dihedral angles. In this case, the overlapping parts of adjacent caps lie close to each other. In practice testing all points in the outer halfspace therefore shouldn't be a problem.

Testing supertriangles for validity results in more bounds for the corresponding patch radius. The center of the spherical cap has to lie on the *center line* of the supertriangle, which is the line passing through the circumcenter and is perpendicular to that triangle. If an input point inside the corresponding halfspace needs to be ϵ-close to the spherical cap, this condition gives an interval of valid cap centers on the center line. If the intersection of all of these intervals together with the half-line defined by the lower bound is nonempty, the supertriangle is valid. Since the lower bound r_i^{min} for proper intersection of caps ensures that the surface is convex, we expect the intersection of these intervals to lie within that valid half-line.

First we test all supertriangles for validity. If there are invalid supertriangles, we add an input vertex to S and update the convex hull incrementally. There are different criteria by which the new vertex can be chosen. Note that refining a supertriangle makes the pyramid condition for adjacent supertriangles stricter, thus a valid supertriangle can become invalid. We test the validity for the newly constructed supertriangles and for all neighbors of new supertriangles. This way, we gradually refine the polytope and the approximating curved surface without having to revalidate the entire structure.

For an invalid supertriangle we have to choose an outlying input point to add to the polytope. We have tried combinations of different measures. The most succesful measure is the distance to the supporting plane, similar to the Douglas-Peucker algorithm for planar curves [31]. That is, for an invalid supertriangle we insert the vertex furthest above the supporting plane of that supertriangle.

Lemma 4.3. *The greedy algorithm terminates.*

Proof. All triangles of the convex hull of P are valid supertriangles, since its three vertices are the only vertices inside the halfspace. Choosing the radius of the spherical cap sufficiently large, we have a proper intersection with the neighboring caps. After adding at most all vertices of P to S we therefore have a valid approximating surface. □

4.5 Results

We have tested the greedy algorithm on different convex point sets. We used point samples on an ellipsoid, a paraboloid and a cylinder, with part of the sample points perturbed (maintaining convexity) by at most a given distance κ. Each set consists of 1000 vertices. The following table shows the number of vertices used in the polytope for different approximation errors ϵ:

data set	$\varepsilon = 0.1$	$\varepsilon = 0.05$	$\varepsilon = 0.01$
cylinder, 10% perturbed with $\kappa = 0.05$	63	125	173
cylinder1, 20% perturbed with $\kappa = 0.08$	107	127	195
ellipsoid, 10% perturbed with $\kappa = 0.5$	90	156	246
ellipsoid1, 20% perturbed with $\kappa = 0.8$	86	146	241
paraboloid, 10% perturbed with $\kappa = 0.05$	163	244	391
paraboloid1, 10% perturbed with $\kappa = 0.05$	143	257	430
paraboloid2, 20% perturbed with $\kappa = 0.08$	95	190	370

All data sets were constructed with generators which construct random points on an ellipsoid, a paraboloid and on a cylinder. The points were perturbed by a routine that constructs from a convex point cloud a convex polyhedron and perturbs a user specified number of vertices of the polyhedron without disturbing the convexity. The difference in the point sets displayed in the table above of *paraboloid* and *paraboloid1* is in the height h and coefficient b ($z = b(x^2+y^2), z \leq h$) used to generate the paraboloids. For *paraboloid* we used $h = 3,\ b = 10$ and for *paraboloid1* and *paraboloid2* we used $h = 2,\ b = 5$. The two data sets for cylinders use as parameters for the height $h = 5$ and for the radius $r = 2.5$. The data sets for ellipsoids use $a = 2,\ b = 2,\ c = 1$, with $x^2/a^2 + y^2/b^2 + z^2/c^2 = 1$. Figure 4.5 is an example for an output of our greedy algorithm for the *ellipsoid* data set.

4.6 Conclusion and future work

As a first step towards approximation with non-planar patches we have presented an algorithm using spherical patches to approximate convex point sets. We have shown that finding the optimal solution for this restricted case is NP hard. Furthermore, this case revealed several difficulties related to non-planar approximation, such as constructing a closed surface and preventing quadratic complexity of the number of patches. These problems have been solved using a greedy algorithm based on an inscribed polytope.

The greedy approach gives rise to various variations in how to build up the polytope. We can choose between locally improving a single supertriangle or

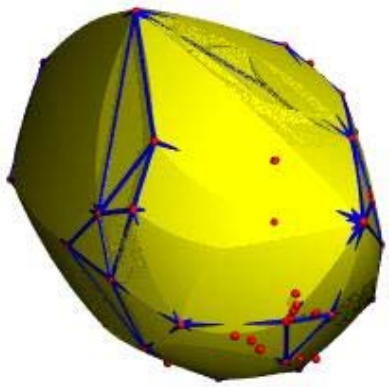


Figure 4.5: Output of the greedy algorithm for the *ellipsoid* data set .

finding the worst invalid supertriangle globally, experiment with different criteria for picking an input point to add to the polytope, or even add multiple input points to the polytope simultaneously. Further experiments are required to examine the various heuristics.

Chapter 5

Approximation by Surface Balls

In this chapter we deal with recovering structural information for a 3-dimensional object that is represented by a sample point cloud. More specifically, given an object $\mathcal{O}$ in 3-space and an r-sample S of its boundary, we want to find an approximating polytope P that uses a subset of the points in S as its vertices and preserves the topology of $\mathcal{O}$. In contrast to the previous chapter, where point clouds in convex position are approximated by spherical patches, we now can deal with non-convex objects. Our goal is, on the one hand, to use as few points of S as possible and, on the other, to get a flexible approximation whose level of detail can be tuned from coarse to fine.

The main support structure we use is an approximation of the object in question with a union of balls. In the context of object simplification, this approach is used for many purposes, e.g. collision detection [50], shape matching [69], and shape interpolation [67], to name a few. Regarding surface reconstruction, approximating objects with balls also plays a major role, see for example the power crust algorithm [9], related work [7, 10, 11] and also [23], naming again only a few.

In our approach, which is similar to work in [23], we build a union of so-called *surface balls*, centered at the points in our r-sample S on the surface $\mathcal{F}$ of $\mathcal{O}$, whose radii adapt to the local feature size of $\mathcal{F}$. The desired approximating polytope P is then extracted from the weighted α-shape[35](a short explanation of α-shapes is given in the Preface) of a carefully chosen subset of these balls. In contrast to [23], where prior knowledge of the local feature size of $\mathcal{F}$ is assumed, we obtain an estimation of this function from the data, by using distances to poles [7] (certain vertices of the Voronoi diagram for S). Using a tailored technique of pruning the surface balls, we obtain a coarse-to-fine approximation of $\mathcal{F}$ by polytopes. This is the first result that uses, from a practical point of view, approximations of local feature size and medial axis to obtain locally adaptive reconstructions of an unknown surface.

The polytopes we construct are topologically correct reconstructions of $\mathcal{F}$. Thus our results differ from existing multi-scale surface reconstruction techniques

in [63, 23, 22, 43] where topological filtering occurs in such cases. Topological filtering implies that small topological features of the surface are ignored in a corase reconstruction, which only accounts for larger features. At the coarsest level, the polytope we obtain is what we call seed polytope, as it provides not only a coarse approximation of $\mathcal{F}$ but also a mapping of the non-used sample points in S to the vertices of the polytope. Such a mapping is needed for incrementally generating approximations of $\mathcal{F}$ based on interpolating subdivision surfaces or Bézier patches. We stress that the intended purpose of the seed polytope is not primarily in approximating $\mathcal{F}$ but rather in serving as a (topologically correct and small) starting structure for subsequent approximations by patches. We thus do not try to keep the approximation error small for the seed polytope itself, and use this additional freedom to keep the polytope small.

Strongly related to the surface reconstruction is the medial axis approximation; we refer to [13] for a recent survey paper on medial axes and their algorithmic construction. In this area, many algorithms are based on unions of balls as well, for example [17, 41, 76]. A variant of our approach [4], now for balls centered at poles instead on sample points, combines with an existing medial axis algorithm for balls [11] to an efficient and stable medial axis approximation algorithm for general objects. It is known that sufficiently dense r-samples lead to topologically correct medial axis approximations; see [10] and, for a result more general than for poles, [12].

Throughout this chapter, let $\mathcal{O}$ denote the original solid object and let $\mathcal{F}$ denote its surface. The following definitions are standard.

Definition 5.1. Medial axis transform:
The medial axis transform *of $\mathcal{O}$ is the (infinite) set of maximal balls that avoid $\mathcal{O}$, where maximality is with respect to inclusion. The set of the centers of these balls forms the* medial axis *of $\mathcal{O}$.*

The surface $\mathcal{F}$ splits the medial axis in an *inner medial axis* and an *outer medial axis*.

Definition 5.2. Local feature size:
The local feature size $\mathrm{lfs}(x)$ *of a point $x \in \mathcal{F}$ is the minimum distance from x to any point on the medial axis of $\mathcal{O}$.*

Definition 5.3. r-sample [7]:
A finite point set $S \subset \mathcal{F}$ is an r-sample *of $\mathcal{F}$ if every point $x \in \mathcal{F}$ has at least one point of S within distance $r \cdot \mathrm{lfs}(x)$.*

From now on, we will assume that S is an r-sample of $\mathcal{F}$ for $r = 0.08$. For each sample point $s \in S$, we define two vertices of the Voronoi diagram of S as the *poles* of s, see [7]: the *inner pole* is the vertex of the Voronoi cell of s farthest away from s and in the interior of $\mathcal{O}$, and the *outer pole* is the farthest one from s and outside $\mathcal{O}$. For the inner pole p of each site s we consider the ball with

center p and radius $\|p-s\|$. We refer to the set of these polar balls as the *(inner) discrete medial axis transform* $\mathrm{DMAT}_{\mathrm{in}}$. Analogously, we generate a set of outer polar balls and denote it by $\mathrm{DMAT}_{\mathrm{out}}$.

Definition 5.4. Discrete medial axis:
The discrete medial axis DM_{in} *(DM_{out}) is the medial axis of the union of polar balls in the sets* $\mathrm{DMAT}_{\mathrm{in}}$ *(*$\mathrm{DMAT}_{\mathrm{out}}$*).*

Definition 5.5. Discrete local feature size:
The discrete local feature size $\widetilde{\mathrm{lfs}}(x)$ *of a point $x \in \mathcal{F}$ is the minimum distance from x to $DM_{\mathrm{in}} \cup DM_{\mathrm{out}}$.*

Definition 5.6. Pole distance:
The pole distance $\hat{D}(x)$ *of a point x is the distance to the nearest pole.*

We will see that $\hat{D}$ is a good estimate of $\widetilde{\mathrm{lfs}}$ (Corollary 5.1), as well as an upper bound on the true local feature size (Lemma 5.5). In practice, $\hat{D}$ is easier to compute than $\widetilde{\mathrm{lfs}}$, and the true local feature size is not computable at all.

The *weighted α-shape* is the dual shape of a union of balls [35]. It is a simplicial complex whose vertices are the centers of the balls, and which is homotopy-equivalent (defined below) to the union of balls. We will refer to the weighted α-shape of $\mathrm{DMAT}_{\mathrm{in}}$ as $\mathcal{A}_{\mathrm{in}}$ and to the one of $\mathrm{DMAT}_{\mathrm{out}}$ as $\mathcal{A}_{\mathrm{out}}$.

Proposition 5.1. *[11] Let $\mathcal{A}_{\mathrm{in}}$ and $\mathcal{A}_{\mathrm{out}}$ be the weighted α-shapes of* $\mathrm{DMAT}_{\mathrm{in}}$,$\mathrm{DMAT}_{\mathrm{out}}$*. Then we have*

$$DM_{\mathrm{in}} \cup DM_{\mathrm{out}} \subseteq \mathcal{A}_{\mathrm{in}} \cup \mathcal{A}_{\mathrm{out}}.$$

If we refer to topological spaces, we focus on $\mathbb{R}^d$ or subsets of $\mathbb{R}^d$. An introduction to the theory of topologycal spaces can be found in [18, Chapter 1].

Definition 5.7. Homeomorphism [36]:
A homeomorphism *is a function $f : \mathbb{X} \to \mathbb{Y}$ between topological spaces that is bijective, continuous and has a continuous inverse. If a homeomorphism exists then $\mathbb{X}$ and $\mathbb{Y}$ are* homeomorphic*.*

Definition 5.8. Homotopy [27]:
A continuous function $g : \mathbb{X} \to \mathbb{Y}$ is homotopic *to another continuous function $h : \mathbb{X} \to \mathbb{Y}$ if there is a continuous function $H : \mathbb{X} \times [0,1] \to \mathbb{Y}$ so that $H(x,0) = g(x)$ and $H(x,1) = h(x)$. The continuous function H is called a* homotopy *between g and h .*

A *homotopy equivalence* is another notion of similarity among topological spaces which is weaker than homeomorphism.

Definition 5.9. Homotopy equivalence [27]:
Two spaces $f : \mathbb{X}$ and $\mathbb{Y}$ are homotopy equivalent *if there exist continuous functions $g : \mathbb{X} \to \mathbb{Y}$ and $h : \mathbb{Y} \to \mathbb{X}$ so that $h \circ g$ is homotopic to the identity function $I_1 : \mathbb{X} \to \mathbb{X}$ and $g \circ h$ is homotopic to the identity function $I_2 : \mathbb{Y} \to \mathbb{Y}$.*

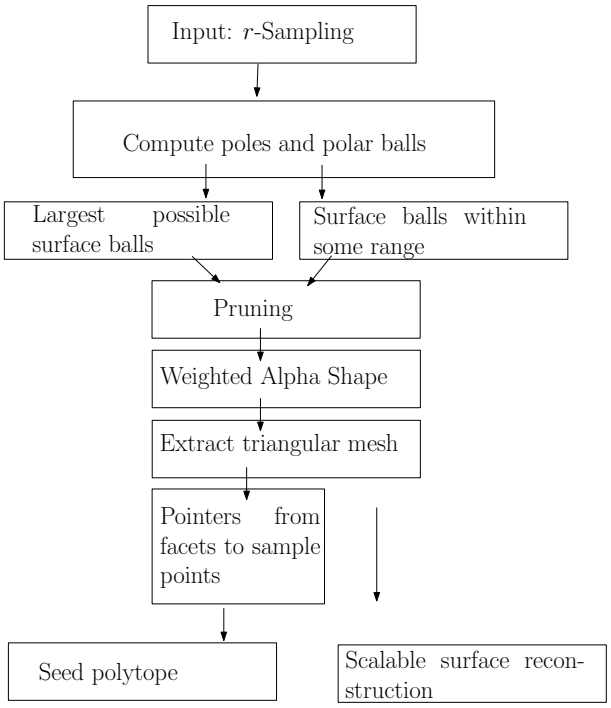


Figure 5.1: Work flow

5.1 Our approach

The main focus in this chapter is on two tasks: Computing a seed polytope, and a scalable surface reconstruction, see also the flowchart in Figure 5.1.

In both cases we start with an r-sample S of the object $\mathcal{O}$ as input and compute from it the two discrete medial axis transforms $\mathrm{DMAT}_{\mathrm{in}}$ and $\mathrm{DMAT}_{\mathrm{out}}$. For seed polytopes and scalable surface reconstruction we use them in order to estimate bounds on the local feature size of the sample points.

5.1.1 The union of surface balls.

A *surface ball* is a ball with center at a sample point $s \in S$. For seed polytopes, our goal is to represent the surface $\mathcal{F}$ of $\mathcal{O}$ in a topological correct way with as few faces as possible. We try to make the surface balls as large as possible, while guaranteeing correct topology of the the union $U(B_{\mathcal{F}})$ of the set $B_{\mathcal{F}}$ of surface balls. A subsequent pruning step will throw away some of these balls whenever the sample is denser than necessary. For surface reconstruction, we will output meshes of scalable complexity. The only modification necessary to reach this goal is to choose surface balls with smaller radii.

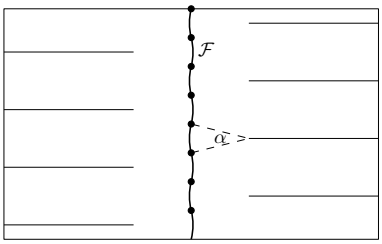


Figure 5.2: A wiggly curve $\mathcal{F}$ with a point sample on a straight line. (Adapted from [10].)

5.1.2 Pruning.

To decide which balls to keep, we solve a combinatorial problem. We (virtually) shrink the balls in $B_{\mathcal{F}}$ and compute a minimal subset $B'_{\mathcal{F}}$ of $B_{\mathcal{F}}$ such that the shrunk balls cover the sample S. This is a *set covering problem*, which is solved by a heuristic. The advantage of this approach is that the selection of the pruned subset proceeds now in a purely combinatorial manner, without regard to geometry and topology. The radii of the shrunk balls are chosen in such a way that covering of S by a subset of shrunk balls guarantees that the original, unshrunk, surface balls cover the surface $\mathcal{F}$, and moreover, their union represents the topology of $\mathcal{F}$ correctly.

5.1.3 The polyhedral approximation.

Finally we compute the weighted α-shape of $B'_{\mathcal{F}}$, which has the same topology as $\mathcal{F}$ and which gives the desired seed polytope. The vertices of the weighted α-shape are points in S, because the centers of the balls in $B'_{\mathcal{F}}$ have been chosen from S. We use the power diagram of $B'_{\mathcal{F}}$ to find out which vertex of the polytope each sample point $s \in S$ belongs to and provide a list of pointers representing this relation. With this mechanism we provide a mapping of the non-used sample points in S to the vertices of the polytope.

5.1.4 Obtaining the local feature size.

A distinguishing feature of our problem setting is that we cannot get a lower estimate on the local feature size. Figure 5.2 shows a section of a curve $\mathcal{F}$ that consists of alternating short circular arcs. The horizontal lines are part of the medial axis. The points of the r-sample S are aligned vertically. By reducing the angle α, such an example can be built for any $r > 0$. The algorithm sees only these samples. Thus, to the algorithm, this input is indistinguishable from a very densely oversampled straight line.

5.2 Technical results

In order to generate adequate sets of polar balls and surface balls (in both cases, the topology must be maintained), we need to derive certain information concerning the local feature size of the sampled object. The present and the subsequent section are devoted to this issue. We obtain several new properties of r-sampled objects for suitable values of r.

Let $\mathcal{M}_{\text{in}}$ and $\mathcal{M}_{\text{out}}$ denote the inner and the outer medial axis of the given object $\mathcal{O}$, respectively. We start by bounding the distance of poles to the respective parts of the medial axis—a result crucial for bounding the radii of surface balls in Section 5.3.

Theorem 5.1. *For an r-sample S, let p be an inner (resp. outer) pole of a sample point $s \in S$, and denote with B_p the inner (outer) polar ball of s, with radius R_p. The distance from p to $\mathcal{M}_{\text{in}}$ ($\mathcal{M}_{\text{out}}$) is at most $O(r) \cdot R_p$.*

In the limit, when the sampling density approaches zero, poles and the medial axis coincide, as has already been shown by Amenta et al. [9, Theorem 35]. In contrast to this result, we give an explicit quantitative analysis in terms of r. Results similar to Theorem 5.1 have been shown (see e. g. [9, Lemma 34], on which Theorem 35 is based, or [15, Proposition 16]). However, we could not use these results, since they hold only when the angle between the two closest surface points to a given point on $\mathcal{M}_{\text{in}}$ ($\mathcal{M}_{\text{out}}$) is not too small, (These points form the *γ-medial axis*.)

Proof. The idea of the proof is to turn the polar ball B_p into a medial ball, while not moving its center too much. The proof is based on several technical lemmas which are given subsequently. We proceed in three steps, see Figure 5.3:

1. While keeping the center of B_p fixed we shrink the radius of B_p until the ball becomes empty, touching the surface $\mathcal{F}$ of $\mathcal{O}$ at some point x_0. By Lemma 5.1 below, the difference Δ_1 between the new radius and the original radius R_p is at most $\Delta_1 = O(r^2) \cdot R_p$.

2. We expand the shrunken ball from the touching point x_0 by moving its center in the direction $\vec{x_0 p}$ until either

   (2a) the ball has the original radius R_p of B_p, or

   (2b) the ball touches the surface at another point. If this occurs we have found a point of $\mathcal{M}_{\text{in}}$ within distance Δ_1, and we are done.

3. In case (2a), we "roll" the new ball B'_p (with radius R_p) on the surface. More precisely, let K_1 be the component of $B_p \cap \mathcal{F}$ which contains x_0. Consider the balls of radius R_p that are tangent to $\mathcal{F}$ in a point of K_1 and lie on the same side of $\mathcal{F}$ as p. The locus of the centers of these balls is

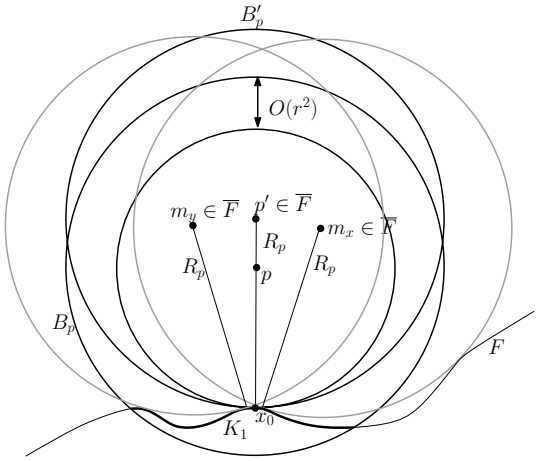


Figure 5.3: After shrinking and expanding the ball B_p we roll the new ball B'_p on K_1 (e.g. the gray ball).

the inner parallel surface $\bar{F}$ of K_1. We claim that the rolling ball touches another point of $\mathcal{F}$, and therefore $\bar{F}$ contains a point of $\mathcal{M}_{\text{in}}$.

We prove this by contradiction. Let us suppose that the ball can roll on K_1 without ever touching a second point of $\mathcal{F}$. K_1 cuts B_p into two parts: B^+ containing p, and the rest B^-. By Lemma 5.4 below, B^+ is completely covered by the tangent balls of K_1. Since by assumption these balls never hit another point of $\mathcal{F}$, it follows that K_1 is the only component of $\mathcal{F} \cap B_p$. Let $s \in B_p$ be the sample point whose pole is p. This point must lie on K_1 and therefore we can roll the empty tangent ball of radius R_p to s. The radius R_M of the medial ball at s is therefore at least R_p. On the other hand, each point of the medial axis is contained in the Voronoi cell of the nearest sample point, therefore $\|p - s\| = R_p \geq R_M$. This implies $R_p = R_M$ and the tangent ball at s has its center on $\mathcal{M}_{\text{in}}$, and we are done. We remark that this last case can actually never arise, since $R_p > R_M$ unless the medial axis branches and the ball touches $\mathcal{F}$ in several points.

We have established that $\bar{F}$ contains a point m_x of $\mathcal{M}_{\text{in}}$ which is the center of a medial ball with radius R_p touching K_1 in x. We know by Lemma 5.3a that the angle $\gamma = \angle m_x x p$ is at most $3r + O(r)$. Thus, $\|p - m_x\| \leq R_p \cdot (3r + O(r^2))$.

□

In the following, we will assume that p is an inner pole. (The situation is symmetric for outer poles.)

Lemma 5.1. *Let p be a pole with polar radius R_p. The surface $\mathcal{F}$ cannot get*

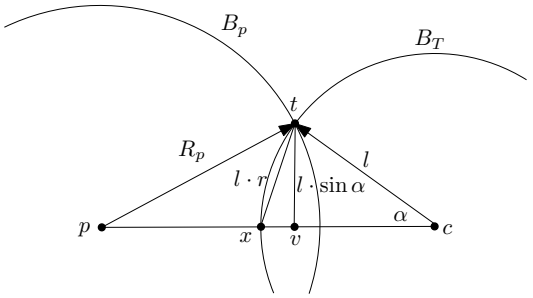


Figure 5.4: Deepest penetration into B_p

closer to p than

$$R_p\Big(\sqrt{1-4(r^2-\tfrac{r^4}{4})}-r^2\Big) \geq R_p\left(1-3r^2-O(r^4)\right).$$

For an r-sample with $r = 0.08$ the distance between the center p of a polar ball with radius R_p and $\mathcal{F}$ is larger than $0.9807 \cdot R_p$.

Proof. Let x be the point on $\mathcal{F}$ closest to p. Let B_T be an empty outer ball tangent to x with center c and radius $l = \mathrm{lfs}(x)$. By the sampling condition, there must be a sample t within distance rl of x. t lies outside the balls B_p and B_T and therefore the distance from x to the circle $\partial B_p \cap \partial B_T$ is at most $r \cdot l$ (see Figure 5.4). Thus, the angle $\alpha = \angle cpt$ is bounded by $\sin\frac{\alpha}{2} \leq \frac{r}{2}$. For fixed l and R_p, the point x is closest to p when α is maximized. We thus analyze the situation for $\sin\frac{\alpha}{2} = \frac{r}{2}$:

$$\sin\alpha = 2\sin\frac{\alpha}{2}\cos\frac{\alpha}{2} \leq 2\cdot\frac{r}{2}\sqrt{1-\tfrac{r^2}{4}} = \sqrt{r^2-\tfrac{r^4}{4}}$$

$$\|v-p\| = \sqrt{R_p^2-(l\cdot\sin\alpha)^2} = \sqrt{R_p^2-l^2\cdot(r^2-\tfrac{r^4}{4})}$$

$$\|v-x\| = \sqrt{(l\cdot r)^2-(l\cdot\sin\alpha)^2} = \sqrt{(l\cdot r)^2-l^2\cdot(r^2-\tfrac{r^4}{4})} = \frac{l\cdot r^2}{2}$$

$$\|x-p\| \geq \|v-p\| - \|v-x\|\sqrt{R_p^2-l^2\cdot(r^2-\tfrac{r^4}{4})} - \frac{l\cdot r^2}{2}$$

The inner polar ball B_p contains a point of $\mathcal{M}_{\text{in}}$ ([9, Corollary 13]), therefore $l \leq 2R_p$. It follows that the distance between p and $\mathcal{F}$ is at least

$$\sqrt{R_p^2-4\cdot R_p^2\cdot(r^2-\tfrac{r^4}{4})} - R_p\cdot r^2 \;=\; R_p\cdot\Big(\sqrt{1-4\cdot(r^2-\tfrac{r^4}{4})}-r^2\Big),$$

as claimed in the lemma. □

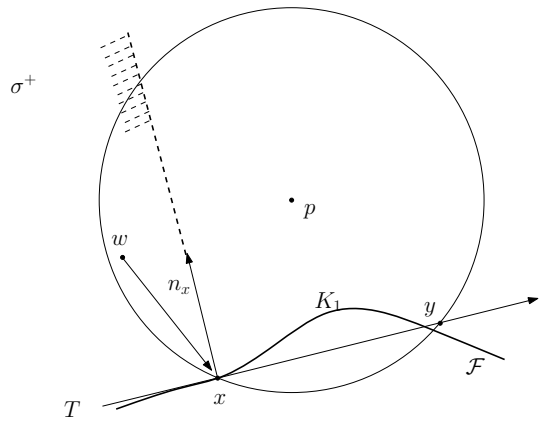


Figure 5.5: The tangent balls of K_1 cover B^+

Lemma 5.2. *Let B_p denote a polar ball with center p. For $r < 0.25$, the normal at a surface point $x \in B_p$ is never perpendicular to the ray $\overrightarrow{px}$.*

Proof. By contradiction: We assume that $\mathrm{lfs}(x) = l$. The points m_1, m_2 are centers of two tangent balls in x of radius l, both balls are empty, one is located inside $\mathcal{F}$ and the other one is outside. The surface must cross the path $m_1 p m_2$ at least once. Assume it crosses $m_1 p$ in some point y. Then the following holds (see Figure 5.6):

$$\begin{aligned} R_p - \|p - m_1\| + \|m_1 - y\| &= R_p - \|y - p\| \\ &= \mathrm{dist}(y, \text{boundary of } B_p) \\ &\leq \mathrm{dist}(y, \text{closest sample}) \leq r \cdot \mathrm{lfs}(y) \\ &\leq r(l + \|x - y\|) \leq r(l + \|x - m_1\| + \|m_1 - y\|) \\ &= r(2l + \|m_1 - y\|). \end{aligned}$$

Therefore,

$$\begin{aligned} \|p - m_1\| - R_p &\geq \|m_1 - y\|\,(1 - r) - 2rl \\ &\geq l(1 - r) - 2rl \geq l(1 - 3r) \end{aligned}$$

and also (according to Lemma 5.1)

$$\|p - m_1\| - l \geq \|p - y\| \geq R_p \left(\sqrt{1 - 4(r^2 - \tfrac{r^4}{4})} - r^2 \right).$$

If $\angle p x m_1$ is a right angle, then $\|p - m_1\|^2 = \|p - x\|^2 + l^2 \leq R_p^2 + l^2$. We have $l > 0, R_p > 0, \|p - m_1\| > 0$, so
$\|p - m_1\|^2 \geq (l(1 - 3r) + R_p)^2$ therefore we get $(l(1 - 3r) + R_p)^2 = R_p^2 + l^2$. For $l < 2R_p$ and $r \leq 0.25$ this leads to a contradiction. □

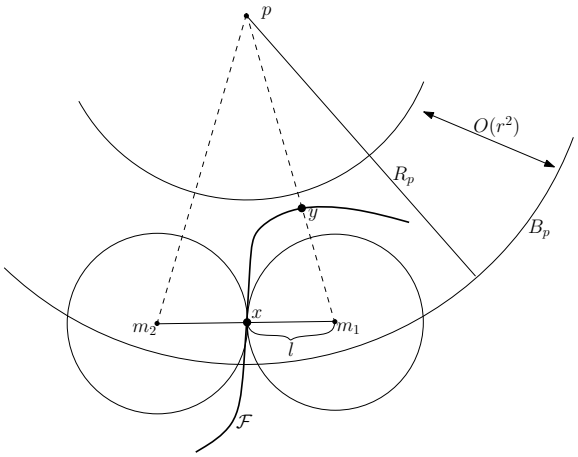


Figure 5.6: An absurd situation; the normal in x is perpendicular to the ray px. The points m_1 and m_2 can lie inside B_p (as in the picture) or outside.

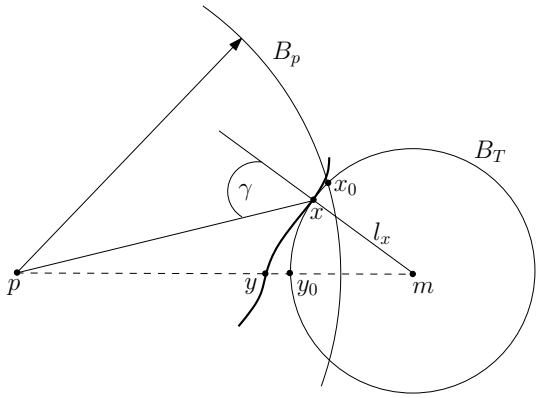


Figure 5.7: Bounding the angle γ. Intersection of the polar ball B_p and the empty tangent ball B_T.

Lemma 5.3. *Let x be a surface point x inside a polar ball B_p with center p.*

a) The angle γ between $\overrightarrow{xp}$ and the surface normal at x is bounded by $3r + O(r^2) = O(r)$.

b) *(The penetration bound) The distance from x to the boundary of B_p is is bounded by $\frac{3}{2}\,\mathrm{lfs}(x)(r^2 + O(r^3))$.*

Part b of the lemma is similar to Lemma 5.1, except that the penetration of the surface point x into the pole ball B_p is measured in terms of $\mathrm{lfs}(x)$, and not in terms of the radius of B_p.

Proof. Since the two parts have the same assumption, we start with calculations which are common to both claims. Consider the empty tangent ball B_T at x of radius $l_x = \mathrm{lfs}(x)$ on the opposite side of p and let m denote its center, see Figure 5.7. For any surface point z inside B_p the ray $\overrightarrow{pz}$ intersects the surface

transversely and never tangentially by Lemma 5.2, therefore the surface patch around x must pass between p and B_T and cannot fold back. In particular, it must intersect the segment $\overline{mp}$ at some point y. The distance from y to the closest sample point s is $\|y-s\| \geq \|y-x_0\|$, where x_0 is the closest point of the circle $B_p \cap B_T$. By the sampling condition we know

$$\|y-x_0\| \leq \|y-s\| \leq r \cdot \mathrm{lfs}(y)$$

and using the Lipschitz condition we get

$$r \cdot \mathrm{lfs}(y) \leq r \cdot (\mathrm{lfs}(x) + \|x-y\|) \leq r \cdot (\mathrm{lfs}(x) + \|y-x_0\|)$$

Thus

$$\|y-x_0\| \leq \frac{r}{1-r} l_x$$

The intersection point $y = \overline{mp} \cap \mathcal{F}$ for which $\|y-x_0\|$ is smallest is y_0 with $\|y_0 - m\| = l_x$,

$$\|y_0 - x_0\| \leq \|y-x_0\| \leq \frac{r}{1-r} l_x.$$

We get for the angle $\alpha = \angle pmx_0$:

$$\alpha = \angle pmx_0 = 2 \arcsin \frac{\|y_0 - x_0\|/2}{l_x} \leq 2 \arcsin \frac{r/2}{1-r}.$$

We now continue with the proof of part (a). A similar assertion [9, Lemma 17] says that the angle at which B_T and B_p intersect is at most $2\arcsin(2r)$, which is less than $\frac{\pi}{2}$ for $r < 0.35$. Therefore, if x varies on the tangent sphere B_T, the largest possible γ (within B_p) is achieved when x is on x_0. From now on we assume $x = x_0$ (see Figure 5.8). We have

$$\gamma = \alpha + \beta. \qquad \sin\beta = \frac{l_x \sin\alpha}{R_p} \leq 2\sin\alpha$$

because $l_x \leq 2R_p$ according to [9, Corollary 13], so

$$\begin{aligned} \beta &\leq \arcsin(2\sin\alpha) \\ \beta + \alpha &\leq 3r + 3r^2 + \tfrac{33}{8}r^3 + \tfrac{51}{8}r^4 + O(r^5) = 3r + O(r^2) \\ \gamma &\leq 3r + O(r^2) = O(r), \end{aligned}$$

as claimed in the lemma.

(b) The distance d from x to B_p is no more than the distance of y_0 to B_p.

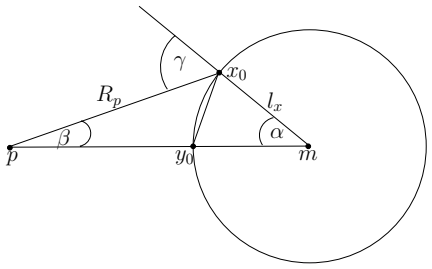


Figure 5.8: Bounding the angle γ.

We can therefore use Figure 5.8:

$$\begin{aligned}
d &= R_p(1-\cos\beta) + l_x(1-\cos\alpha) \\
&= R_p(1-\sqrt{1-\sin^2\beta}) + l_x(1-\sqrt{1-\sin^2\alpha}) \\
&= R_p(1-\sqrt{1-\sin^2\alpha\cdot l_x^2/R_p^2}) + l_x(1-\sqrt{1-\sin^2\alpha}) \\
&= \tfrac{1}{2}\cdot[R_p(\sin^2\alpha\cdot l_x^2/R_p^2 + O(\alpha^4)) + l_x(\sin^2\alpha + O(\alpha^4))] \\
&= \tfrac{1}{2}\cdot[(\sin^2\alpha\cdot l_x(l_x/R_p) + O(\alpha^4)) + l_x(\sin^2\alpha + O(\alpha^4))] \\
&\leq \tfrac{1}{2}\cdot[(\sin^2\alpha\cdot 2l_x + O(\alpha^4)) + l_x(\sin^2\alpha + O(\alpha^4))] \\
&= l_x\cdot(\tfrac{3}{2}r^2 + 3r^3 + O(r^4)).
\end{aligned}$$

□

To complete the proof of Theorem 5.1, we still need to show that the tangent balls of K_1 cover all parts of B^+. Recall that K_1 cuts B_p in two parts: B^+ containing p, and the rest B^-.

Lemma 5.4. *The tangent balls of K_1 completely cover B^+.*

Proof. Let $w \in B^+$ and let x be the closest point of K_1. We claim that the tangent ball at x covers w. If x lies in the interior of K_1, then wx is perpendicular to $\mathcal{F}$, and the claim is obvious. Let us assume that x is at the boundary of K_1, that is $B_p \cap \mathcal{F}$ (see Figure 5.5). Assume that the surface normal n_x does not go through p; otherwise it is obvious that w is covered. Consider the plane σ through n_x and through the point p. Figure 5.5 shows the projection on this plane. Locally around x, $\mathcal{F}$ is approximated by the tangent plane T and $B_p \cap \mathcal{F}$ is the halfspace of T that projects onto the ray xy in Figure 5.5. It follows that x can only be the point of K_1 closest to w, if w lies in the plane σ and in the closed halfplane σ^+ of σ which is bounded by n_x and does not contain p. □

5.3 Construction of balls

5.3.1 Polar balls

For the set $\mathrm{DMAT}_{\mathrm{in}}$ of inner polar balls, it is well known [10] that the union of the balls in this set is homeomorphic to the original object $\mathcal{O}$. Recall that each ball in $\mathrm{DMAT}_{\mathrm{in}}$ is the circumball of a Delaunay tetrahedron and therefore has at least four points of S on its boundary and no such point in its interior. From $\mathrm{DMAT}_{\mathrm{in}}$ we generate a set $\mathrm{DMAT}'_{\mathrm{in}}$ of slightly enlarged balls which are still centered on S. Such a ball typically covers tens or even hundreds of points of S. In a subsequent set covering step, this redundancy in covering will be eliminated, and thereby only a small and stable subset of $\mathrm{DMAT}'_{\mathrm{in}}$ will be kept. We have to ensure, for the goal of topologically correct medial axis approximation, that the union of $\mathrm{DMAT}_{\mathrm{in}}$ and the union of $\mathrm{DMAT}'_{\mathrm{in}}$ are topologically equivalent. Using the lower bound on the discrete local feature size of sample points developed in Lemma 5.8 below, it is easy to check whether $\mathrm{DMAT}'_{\mathrm{in}} \cap \mathcal{A}_{\mathrm{out}} = \emptyset$.

5.3.2 Surface balls

In order to maintain correct topology of the piecewise linear surface reconstruction, the surface balls we generate have to be large enough such that their union does not only cover S but also $\mathcal{F}$ and, on the other hand, these balls avoid the medial axis of the union of the balls in $\mathrm{DMAT}_{\mathrm{in}}$ and $\mathrm{DMAT}_{\mathrm{out}}$. The above restrictions limit the possible radii to a certain range. Maximizing the radii within this range will lead to a coarse result (which is desirable for seed polytopes), while minimizing the radii of the surface balls will lead to a faithful and detailed representation of the object. The choice of the radii determines the degree by which the surface balls are pruned in a subsequent set covering step.

Lower bound on the radii

To ensure that $\mathcal{F}$ is completely covered by surface balls we choose the radii of the surface balls such that they cover at least the intersection of their site's Voronoi cells with $\mathcal{F}$. For a point s in an r-sample, this intersection is covered by a sphere around s whose radius is $\rho \geq \frac{r}{1-r} \cdot \mathrm{lfs}(s)$, see [7], and so the surface balls need to have at least that radius. As $\mathrm{lfs}(s)$ is unknown, we need to estimate it in terms of the distance $\hat{D}(s)$ between s and the nearest among the poles of all sample points. Using Lemma 5.5 below, we get

$$\mathrm{lfs}(s) \leq 1.2802 \cdot \hat{D}(s)$$

and so we must choose the radius ρ of a surface ball around s to be at least

$$\rho \geq \tfrac{r}{1-r} \cdot 1.2802 \cdot \hat{D}(s).$$

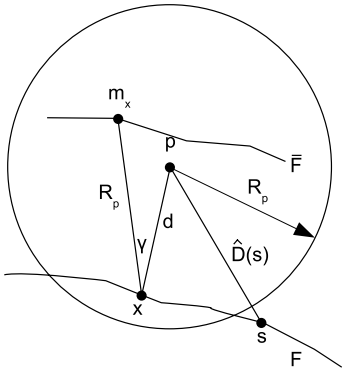


Figure 5.9: Distance from pole p to the medial axis point m_x

The distance $\hat{D}(s)$ can be calculated relatively easily using a spatial search structure.

Lemma 5.5. *Let $s \in S$ be a point of an r-sample S with $r \leq 0.08$, and let $\hat{D}(s) = \|s - p\|$ denote its distance to the nearest pole p. Then*

$$\text{lfs}(s) \leq 1.2802 \cdot \hat{D}(s).$$

Proof. The local feature size of s cannot be larger than $\hat{D}(s)$ plus the distance from p to the medial axis. To bound the latter distance for a specific value of r, we revisit the cases developed in Theorem 5.1 (and we use the notation introduced there). If case (2a) occurs we know that $\bar{F}$ contains a point $m_x \in \mathcal{M}_{\text{in}}$ ($\mathcal{M}_{\text{out}}$); see Figure 5.9. By Lemma 5.3a, the maximum angle between the touching point $x \in K_1$ of the medial ball centered at m_x and p is $\gamma = \angle m_x x p < 14.99°$ if $r \leq 0.08$. By Lemma 5.1,

$$d = \|x - p\| \geq \left(\sqrt{1 - 4(r^2 - \tfrac{r^4}{4})} - r^2\right) \cdot R_p > 0.9807 \cdot R_p.$$

Therefore

$$\|p - m_x\| \leq 2 \cdot R_p \cdot \sin(\frac{\gamma}{2}) + (1 - 0.9807) R_p < 0.2802 \cdot R_p$$

which is at most $0.2802 \cdot \hat{D}(s)$ because s lies outside the polar ball centered at p. Otherwise, case (2b) occurs and by Lemma 5.1, p is not farther from $\mathcal{M}_{\text{in}}$ ($\mathcal{M}_{\text{out}}$) than

$$R_p \cdot (1 - \sqrt{1 - 4 \cdot (r^2 - \tfrac{r^4}{4})} + r^2) < 0.0193 \cdot R_p.$$

The lemma follows. □

Upper bound on the radii

To prevent surface balls from "different" parts of $\mathcal{F}$ from intersecting we want to ensure that they don't reach the discrete medial axis DM_{in} (resp. DM_{out}).

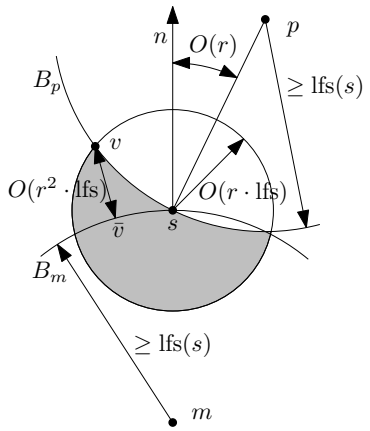


Figure 5.10: A point v that is not covered by the polar ball must lie close to the surface.

Thus, the discrete local feature size $\widetilde{\text{lfs}}(s)$ is an upper bound on the radius that we can use. We will replace $\widetilde{\text{lfs}}(s)$ by a smaller value, that is easier to compute, see Proposition 5.1.

Consequently, the minimum distance from s to any of the two weighted α-shapes is a lower bound on $\widetilde{\text{lfs}}(s)$. Computing $\mathcal{A}_{\text{in}}$ and $\mathcal{A}_{\text{out}}$ and determining the minimum distance directly would consume too much time and memory, however. We show how to estimate this distance, again using the distance $\hat{D}(s)$ to the nearest pole to s.

Lemma 5.6. *Let s be a sample point, and let v be a point with the following properties*

- *v lies in the Voronoi cell of s.*
- *v is not in the interior of the polar ball around the pole p of s that lies on the same side of $\mathcal{F}$ as v.*

Then

(a) $\|v - s\| = O(r) \cdot \text{lfs}(s)$. *In particular, for $r = 0.08$, the distance to s is at most $0.123 \cdot \text{lfs}(s)$.*

(b) *The distance from v to the closest point $\bar{v}$ on the surface is $O(r^2 \,\text{lfs}(s)) = O(r^2 \,\text{lfs}(\bar{v}))$. For $r = 0.08$, the distance $\|v - \bar{v}\|$ is at most $0.0355 \cdot \text{lfs}(s) \leq 0.0424 \cdot \text{lfs}(\bar{v})$.*

Proof. (of Lemma 5.6)

We perform the calculation for $r = 0.08$, and only indicate the asymptotic dependence on r. We will first show part (a).

$$\|v - s\| \leq 0.123 \cdot \text{lfs}(s) = O(r \,\text{lfs}(s)).$$

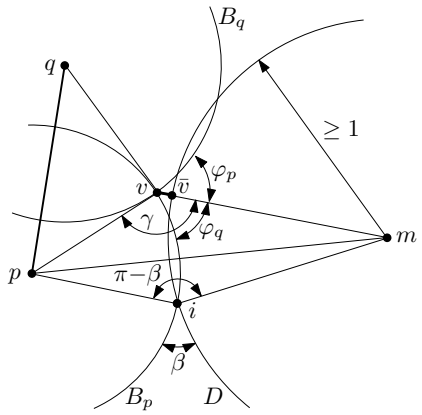


Figure 5.11: Schematic figure of an intersection of two polar balls such that their intersection point v is not covered by the union of polar balls.

Let p be the pole of s on the same side of the surface as v. If $\|v-s\| > kr\cdot\mathrm{lfs}(s)$ for $k = 1.536$, the angle between sv and the surface normal is at most

$$\arcsin\frac{1}{k(1-r)} + \arcsin\frac{r}{1-r} < 47.2^\circ,$$

see [7, Lemma 4]. Similarly, the angle between the normal and sp is at most $2\arcsin\frac{r}{1-r} < 12.8^\circ$. In total the angle vsp is less than 60°. Since $\|v-s\| \leq \|p-s\|$, by the definition of the pole, it follows that v must be contained in the polar ball around p, whose radius is $\|p-s\|$, a contradiction. We thus conclude that v is contained in a ball of radius

$$kr\cdot\mathrm{lfs}(s) \leq 0.123\cdot\mathrm{lfs}(s) \quad (= O(r\,\mathrm{lfs}(s)))$$

around s. Since v avoids the polar ball B_p around p, it lies in the shaded region indicated in Figure 5.10. The direction sp of the polar ball deviates at most $2\arcsin\frac{r}{1-r} < 12.8^\circ$ $(= O(r))$ from the normal direction n at s. Thus the "highest" possible position of v is as indicated in the figure. We know that the surface must pass above the opposite medial ball P_m of s, and thus we can estimate the distance from v to the surface and prove (b). A straightforward calculation gives the bound $\|v-\bar{v}\| \leq 0.0355\,\mathrm{lfs}(s)$ $(= O(r^2\,\mathrm{lfs}(s)))$. By the Lipschitz condition,

$$0.0355\,\mathrm{lfs}(s) \leq \tfrac{0.0355}{1-0.123-0.0355}\,\mathrm{lfs}(\bar{v}) \leq 0.0424\cdot\mathrm{lfs}(\bar{v})$$

is obtained. □

Lemma 5.7. *Let pq be an edge of the weighted α-shape $\mathcal{A}_{\mathrm{in}}$ ($\mathcal{A}_{\mathrm{out}}$). Then the exterior angle of intersection between the polar balls B_q, B_p around p and q is at least* 120°.

Proof. Since pq is an edge of the weighted α-shape, there is a point v on the intersection of the boundaries of the two polar balls B_p and B_q which is not covered by any other polar ball, see Figure 5.11. Therefore, the neighborhood of v contains points outside all polar balls and, by Lemma 5.6.

v is close to $\mathcal{F}$: For the closest surface point $\bar{v}$ we have

$$d = \|v - \bar{v}\| \leq 0.0424 \cdot \mathrm{lfs}(\bar{v}).$$

Without loss of generality, we assume $\mathrm{lfs}(\bar{v}) = 1$. Consider the medial ball B of $\bar{v}$ on the opposite site, with center m and radius $\|\bar{v} - m\| \leq \mathrm{lfs}(\bar{v}) = 1$. By [9, Lemma 17], a polar ball B_p or B_q intersects a medial ball D on the opposite site at angle $\beta \leq 2 \arcsin 2r$. Let us focus on one ball B_p and the angle ϕ_p between this ball and the surface normal vm. The other ball is treated in the same way, and the total exterior angle is then $\phi_p + \phi_q$.

We have $\phi_p = \gamma - \pi$, where $\gamma = \angle pvm$. To get an upper bound on ϕ_p (or on γ), let us fix the angle γ and try to find circles B_p and D that are consistent with this situation. We have the following constraints:

(i) $1 = \mathrm{lfs}(\bar{v}) \geq \|\bar{v} - m\|$;

(ii) $d := \|v - \bar{v}\| \leq 0.0424 \cdot \mathrm{lfs}(\bar{v}) \leq 0.0424$;

(iii) The intersection angle between B_p and D is $\beta \leq 2 \arcsin 2r$.

This gives us a distance $\|c - v\| = 1 + d$, using the triangle inequality we get $\|q - v\| = 1 - d$. For the triangle qcv only the segment qc is of unknown length. We consider also a second triangle, formed by the points q, c and one intersection point i of the medial ball with the polar ball B_q. Again only the distance of the segment qc is unknown. From the triangles we get the following equations:

$$\cos\beta = \tfrac{1+(1-d)^2-\|c-q\|^2}{2(1-d)}, \qquad \cos\gamma = \tfrac{(1+d)^2+(1-d)^2-\|c-v\|^2}{2(1-d)(1+d)},$$

for $\beta = \angle cvq = \pi - \beta = \pi - 2 \arcsin 2r$, $\gamma = \angle qic$, $d = 0.0355$. Solving these equations for γ gives an angle $\varphi = 2 \cdot (\gamma - \pi/2) > 120°$. □

Based on the preceding lemmas, it is possible to derive the following bound on $\widetilde{\mathrm{lfs}}(s)$.

Lemma 5.8. *If m is a point on an edge pq of* $\mathrm{DMAT}_{\mathrm{in}}$ *(or in a triangle pqr of* $\mathrm{DMAT}_{\mathrm{in}}$*) and v is outside or on the boundary of* $U(\mathrm{DMAT}_{\mathrm{in}})$ *then*

$$\|m - v\| \geq 0.817 \cdot \min\{\|p - v\|, \|q - v\|\},$$

(or $\|m - v\| \geq 0.817 \cdot \min\{\|p - v\|, \|q - v\|, \|r - v\|\}$*, respectively).*

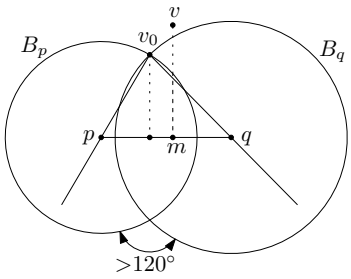


Figure 5.12: The distance from the sample point s to the weighted α-shape

Proof. We first consider the case when m lies on an *edge* pq,as illustrated in Figure 5.12. Let m' be the point on pq that is closest to v. If m' is one of the endpoints p or q, we are done:

$$\|m - v\| \geq \|m' - v\| = \min\{\|p - v\|, \|q - v\|\}.$$

Otherwise we know that $m'-v$ is perpendicular to pq. We know from Lemma 5.7 that the intersection of the two polar balls B_p and B_q cannot be too thin: their angle of intersection is at least 120°. For fixed balls B_p and B_q, the angles and hence the ratios are minimized when s lies on the intersection between the balls (the point v_0 in the figure).

Now keeping v_0 fixed at the intersection and considering a variation of the balls B_p and B_q, maintaining $\min\{\|v - p\|, \|v - q\|\}$, it is clear that the distance from v to the edge pq is minimized when the angle $\angle pvq$ is at its upper bound of 60° and the two distances are equal: $\|v - p\| = \|v - q\|$. Then the ratio $\|v - v\|/\|v - p\| = \cos 30° > 0.866$.

Now consider the case when m lies in a *triangle* pqr. If the point m' on pqr that is closest to v lies on an edge or at a vertex of the triangle, we have reduced the problem to the previous case. Otherwise we know that $m'-v$ is perpendicular to pqr. The remaining argument is similar as in the case of an edge: The extreme situation is a triangular pyramid with equal angles $\angle pvq = \angle qvr = \angle rvp = 60°$ at the apex m and equal sides $\|p-v\| = \|q-v\| = \|r-v\|$. The ratio between the height of this pyramid and the length $\|p - v\|$ is $\sqrt{(1 + 2\cos 60°)/3} > 0.817$. □

Corollary 5.1. *Let $s \in S$ be a sample point, and let $\hat{D}(s)$ be its distance to the nearest pole. Then*

$$\hat{D}(s) \geq \widetilde{\text{lfs}}(s) \geq 0.817 \cdot \hat{D}(s).$$

Proof. Since the poles are part of the discrete medial axis, the inequality $\widetilde{\text{lfs}}(s) \leq D(s)$ is obvious. For the other direction, we bound $\widetilde{\text{lfs}}$ by the distance from v to the weighted α-shape $\mathcal{A}$ of the polar balls, which contains the discrete medial axis. The proof of the lower bound on the ratio

$$\frac{\widetilde{\text{lfs}}(v)}{D} = \frac{\|v - m\|}{D} \geq \max\left\{\frac{\|v - m\|}{\|v - p\|}, \frac{\|v - m\|}{\|v - q\|}\right\},$$

follows from Lemma 5.8. □

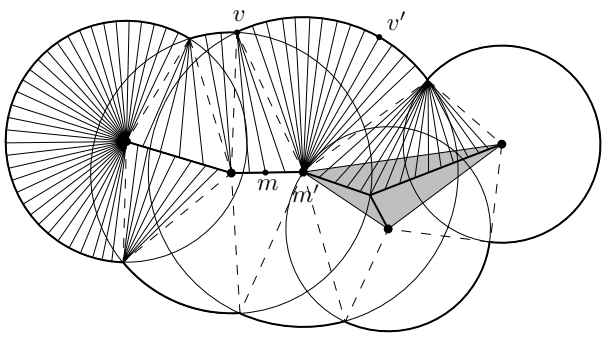


Figure 5.13: Part of the fibration which is used to show isotopy. The shaded area is the weighted α-shape.

5.3.3 Topological Correctness

To show that the union $U(B_F)$ of surface balls is homotopy-equivalent to the surface $\mathcal{F}$, we follow the standard approach of using a fibration (a partition of $U(B_F)$ into a continuous family of curves, each intersecting $\mathcal{F}$ in a single point) and moving the boundaries of $U(B_F)$ along the fibers towards F.

The usual fibration by surface normals does not work since the medial axis might be closer than it appears from looking at the sample points, see Figure 5.2. Instead we use the fibers of the union $U(\mathrm{DMAT_{in}})$ of all polar balls. It is known that this union is homotopy-equivalent to $\mathcal{O}$, and its boundary is homotopy-equivalent to $\mathcal{F}$ [10].

The boundary of the union $U(\mathrm{DMAT_{in}})$ is not smooth, but still, it is in a certain sense "smooth from the inside" (it has no convex edges or vertices) and has therefore a reasonable fibration connecting the boundary to its inner medial axis $\mathrm{DMAT_{in}}$, see Figure 5.13. We concentrate on the inner discrete medial axis $\mathrm{DMAT_{in}}$; the outer discrete medial axis $\mathrm{DMAT_{out}}$ is treated analogously. The fibers are line segments that partition $U(\mathrm{DMAT_{in}}) \setminus DM_{\mathrm{in}}$, and they run from a surface point v on the boundary to a point m on the inner discrete medial axis DM_{in}. In three dimensions, there are three types of fibers: from a point v on a spherical patch of the boundary to a vertex m of the medial axis; from a point v on a circular edge formed as the intersection of two spheres to a point m on an edge of the medial axis; and from a vertex v of the boundary, formed as the intersection of three (or more) spheres to a point m on a face of the medial axis. Our proof treats all three cases uniformly.

We take the radius of the surface balls as $\rho\hat{D}(s)$ where the factor ρ can be chosen in the interval

$$\rho_{\min} = 0.24 \leq \rho \leq \rho_{\max} = 0.56. \tag{5.1}$$

The upper bound ensures that the surface balls do not intersect the discrete medial axis, and the lower bound ensures that they are large enough to cover the surface completely. The bounds are stricter that would be required to reach

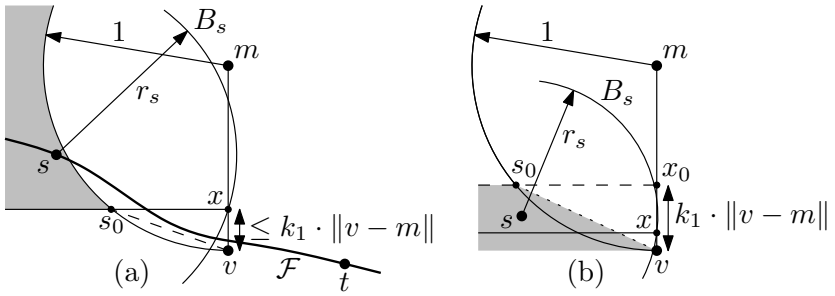


Figure 5.14: A ball B_s that intersects the fiber vm improperly

only these two goals, since we also want to ensure topological correctness of the union $U(B_F)$ of surface balls:

Lemma 5.9. *If ρ is chosen in the interval* (5.1)*, every fiber from a point v on the boundary of $U(\mathrm{DMAT}_{\mathrm{in}})$ to a point m on the medial axis of $U(\mathrm{DMAT}_{\mathrm{in}})$ starts in the union $U(B_F)$ of surface balls and intersects the boundary of $U(B_F)$ precisely once.*

The lemma implies that the boundary of $U(B_F)$ can be continuously deformed along the fibers into the boundary of $U(\mathrm{DMAT}_{\mathrm{in}})$, and thus the two boundaries are homotopy-equivalent. The boundary of $U(\mathrm{DMAT}_{\mathrm{in}})$ is already known to be homotopy-equivalent to $\mathcal{F}$, and thus, the correct topology is established.

Proof. For simplicity we prove the bound for $\rho = 0.3$. The calculation for general ρ is slightly more involved.

Let B_s be a surface ball around a sample point s such that the segment vm enters B_s in a point x, see Figure 5.14a. We will show that this does not lead to a violation of the lemma, because the segment vx is covered by the union of surface balls. We assume without loss of generality that vm is vertical and $\|m - v\| = 1$. We first show that x must have distance $\|x - v\| \leq k_1$ for $k_1 = 0.074$.

Suppose that this is not true. The medial ball of radius 1 around m is inside the union of balls, and hence it does not contain s: $\|s - m\| \geq 1$. We claim that this implies

$$\|s - x\| > 0.37 \cdot \|s - m\|. \tag{5.2}$$

We know that s must lie outside the ball of radius 1 around m; s must also lie above the horizontal line through x. Thus, s is restricted to the shaded area in the figure. The ratio $\|s - x\|/\|s - m\|$ is minimized when x is as low as possible ($\|x - v\| = k_1$) and s is at the lower right corner s_0 of this area. Here we have $\|s - x\|^2 + (1 - k_1)^2 = 1$, from which one can compute $\|s - x\|/\|s - m\| = \|s - x\| > 0.37$.

On the other hand, since $m \in \mathrm{DMAT}_{\mathrm{in}} \subseteq \mathcal{A}_{\mathrm{in}}$, we have by definition $\|s - m\| \geq \widetilde{\mathrm{lfs}}(s) \geq 0.817\hat{D}(s)$, by Lemma 5.8. Thus, the radius r_s of B_s is $r_s = \|s - x\| \leq \rho\hat{D}(s) \leq \rho/0.817 \cdot \|s - m\| < 0.368 \cdot \|s - m\|$, contradicting (5.2).

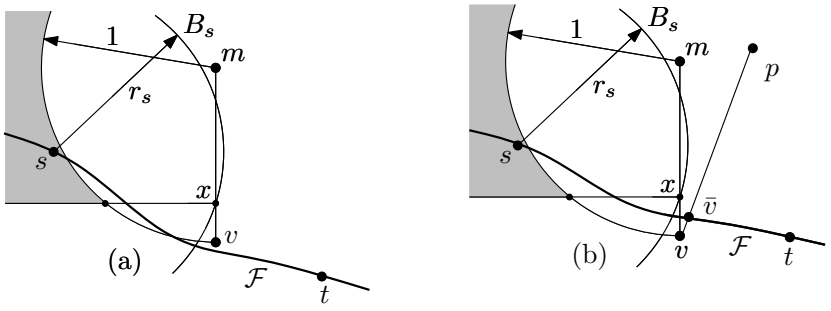


Figure 5.15: A ball B_s that intersects the fiber vm improperly, v lies either inside $\mathcal{F}$ (a) or outside $\mathcal{F}$ (b)

Let us denote the extreme positions of s and x in the above analysis by s_0 and x_0.We have established that x and s lie below horizontal line s_0x_0, see see Figure 5.14b. For an arbitrary x and s we now claim

$$\frac{\|s-x\|}{\|x-v\|} \geq \frac{\|s_0-x_0\|}{\|x_0-v\|} \geq 5. \tag{5.3}$$

We know that s must always lie higher than x, For a fixed point x, we can rotate s around x until it lies at the same height as x, without changing the above ratio, So we can assume that s and x lie at the same height, with $\|x-v\| \leq k_1$. The sample s cannot lie in the polar ball around m, and in particular, s must lie below the dotted line segment. The claim (5.3) follows.

Now to complete the proof we will show that the segment vx is covered by a surface ball, namely by the ball around the surface sample t closest to v. We are done if we can show that the radius r_t of this ball is at least $\|t-v\| + \|v-x\|$:

$$r_t = \rho\hat{D}(t) \geq \|t-v\| + \|v-x\| \tag{5.4}$$

This implies that $r_t \geq \|t-v\|$ and $r_t \geq \|t-x\|$ (by the triangle inequality), and thus ensures that the whole segment vx is covered. It establishes also that the starting point v of the fiber is covered, irrespective of whether another ball B_s intersects vm "in an improper way".

First we show that there is a sample point t with

$$\|t-v\| \leq 0.123 \cdot \mathrm{lfs}(t) \tag{5.5}$$

We distinguish two cases:

(a) v lies inside $\mathcal{F}$ (on the same side as m), see Figure 5.15(a). Let t be the sample point closest to v. The point v satisfies the assumptions of Lemma 5.6 with respect to t: By definition, v lies in the Voronoi cell of t. Moreover, v lies in none of the polar balls around the vertices of $\mathrm{DMAT}_{\mathrm{in}}$. Thus, by Lemma 5.6a, $\|t-v\| \leq 0.123 \cdot \mathrm{lfs}(t)$.

(b) v lies outside $\mathcal{F}$, see Figure 5.15(b). By Lemma 5.8, there is a pole p in $\mathrm{DMAT}_{\mathrm{in}}$ such that

$$\|p-v\| \leq \frac{1}{0.817} \cdot \|m-v\| \leq 1.224 \cdot \|m-v\|$$

The segment vp must intersect $\mathcal{F}$ in some point $\bar{v}$. Lemma 5.3b limits the penetration of the surface point $\bar{v}$ into the ball B_p:

$$\|\bar{v} - v\| \leq (3/2 \cdot r^2 + O(r^3)) \cdot \mathrm{lfs}(\bar{v}).$$

In particular, for $r = 0.08$,

$$\|\bar{v} - v\| \leq 0.0114 \cdot \mathrm{lfs}(\bar{v}).$$

The nearest sample point t from $\bar{v}$ is less than $r \cdot \mathrm{lfs}(t)$ away:

$$\|\bar{v} - t\| \leq r \cdot \mathrm{lfs}(t)$$

The Lipschitz condition yields

$$\mathrm{lfs}(\bar{v}) \leq \mathrm{lfs}(t) + \|\bar{v} - t\| \leq (1 + r) \cdot \mathrm{lfs}(t).$$

Therefore we get:

$$\begin{aligned}
\|t - v\| &\leq \|v - \bar{v}\| + \|\bar{v} - t\| \\
&\leq 0.0114 \cdot \mathrm{lfs}(\bar{v}) + r \cdot \mathrm{lfs}(t) \\
&\leq 0.0114 \cdot (1 + r)\,\mathrm{lfs}(t) + r \cdot \mathrm{lfs}(t) \\
&\leq 0.093\,\mathrm{lfs}(t) \leq 0.123\,\mathrm{lfs}(t)
\end{aligned}$$

proving (5.5).

We have, by Lipschitz continuity, and using (5.3),

$$\begin{aligned}
\hat{D}(t) &\geq \hat{D}(s) - \|s - x\| - \|x - v\| - \|v - t\| \\
&\geq \|s - x\|/\rho - \|s - x\| - \|x - v\| - \|v - t\| \\
&= \|s - x\|(1/\rho - 1) - \|x - v\| - \|v - t\| \\
&\geq 5(1/\rho - 1)\|x - v\| - \|x - v\| - \|v - t\| \\
&= [5(1/\rho - 1) - 1] \cdot \|x - v\| - \|v - t\| \\
&> 10.6 \cdot \|x - v\| - \|v - t\| \qquad (5.6)
\end{aligned}$$

By (5.5) and Lemma 5.5, we have $\|v - t\| \leq 0.123 \cdot \mathrm{lfs}(t) \leq 0.123 \cdot 1.2802 \cdot \hat{D}(t) < 0.1575\hat{D}(t)$ and hence

$$\hat{D}(t) > 6.3 \cdot \|v - t\| \qquad (5.7)$$

Multiplying (5.6) by 0.095, (5.7) by 0.175, and adding them together yields

$$0.27\hat{D}(t) \geq \|x - v\| + \|v - t\|, \qquad (5.8)$$

implying (5.4). □

5.4 Pruning by set covering

If we have a sample that is much denser than required by our conditions, we will get a correct "surface reconstruction", but we would like to obtain a coarser approximation to reduce the data, while maintaining topological correctness. We will therefore only use a subset of the surface balls.

We establish a condition that is easy to check and guarantees the correct topology: As before, we use balls of radius $\rho\hat{D}(u)$ around surface points u; for each ball we also consider a shrunk copy of radius $\bar{\rho}\hat{D}(u)$, where $\bar{\rho} = 0.03 < \rho$. We can then prove the following statement.

Theorem 5.2. *If the shrunk balls around the points u of a subset $S' \subseteq S$ cover all sample points S, then the union of the original balls (of radius $\rho\hat{D}(u)$) around these points is homotopy-equivalent to $\mathcal{F}$.*

Proof. The proof proceeds via the statement of Lemma 5.9. In that proof, we have established the existence of a sample point t that is close enough to v such that the ball around t covers the segment vx. This is extended to the present setting as follows: we can now no longer be sure that the ball around t is used, but there must be a (shrunk) ball around some sample point u that covers t. Then the (original) ball around u is large enough to guarantee that it reaches vx.

We know, by the pruning condition, that the covering contains a ball of radius $\rho\hat{D}(u)$ around a sample point u such that the shrunk ball with radius $\bar{\rho}\hat{D}(u)$ covers t:

$$\|u - t\| \leq \bar{\rho}\hat{D}(u)$$

From this, together with the above bound (5.8) on $\|t - x\|$, we obtain

$$\|u - x\| \leq \|u - t\| + \|t - x\| \leq \bar{\rho}\hat{D}(u) + (\rho - \bar{\rho})\hat{D}(u) = \rho\hat{D}(u),$$

and thus the ball B_u covers x.

The same calculation shows that B_u covers v, and hence the whole segment vx.

□

We try to select a minimum subset of surface balls whose shrunk copies cover the whole sample. This is an instance of the (in general NP-hard) set covering problem.

In practice, these problems can nevertheless be solved efficiently.

In [5] and [4]

a combination of exact and heuristic methods is described which yields not only an approximate solution but also a lower bound on the optimal solution, and in our setting the gap between them is typically quite small.

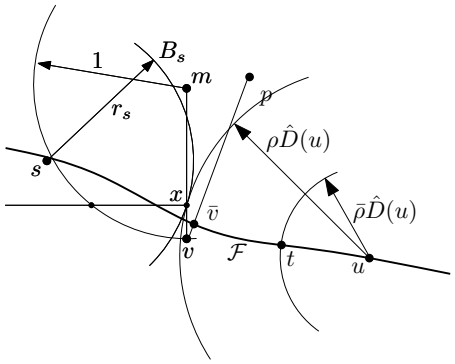


Figure 5.16: The segment vm is covered by the enlarged ball around u.

To get the input data for the set covering problem, the information about the sample points covered by each ball, we use a simple spatial search structure, e.g. a kd-tree.

The lemma remains true if the shrinking factor 0.03 is replaced by a smaller number. This parameter allows us to scale the algorithm to different levels of coarseness or refinement of the approximation. If the shrinking factor approaches 0, each shrunk ball will contain no sample points except its center, and thus the full sample will be used.

The small radius $0.03 \cdot \hat{D}$ that we have proved may not seem very impressive, but it must be seen in relation with the sampling constant $r = 0.08$. Thus, balls will start to be eliminated as soon at the actual sampling density exceeds the required minimum by a factor of about 4–5 (in terms of the sampling radius).

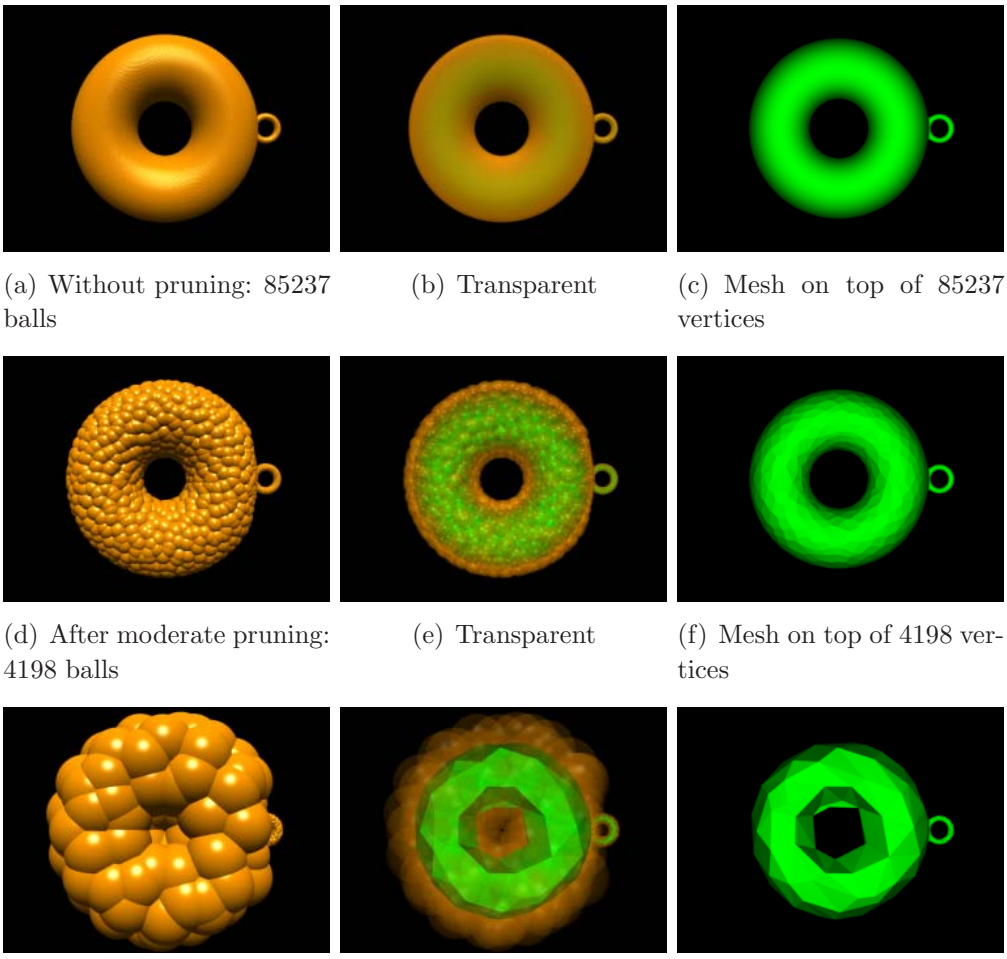
The same approach works for approximating the medial axis. Here we start with an enlarged set of *polar balls* DMAT'_{in}, and produce an (almost) minimum subset DMAT''_{in} whose union covers S.

5.5 Experimental data

The example included shows the output produced by Bernhard Kornberger's implementation of our algorithm for surface reconstruction.

Figure 5.17 illustrates how different choices of radii for surface balls lead to different levels of detail in the approximating polyhedral surface mesh. The initial point cloud for this 'double torus' model consists of 85237 points. Due to the effect of pruning, the mesh for the big ring is more and more coarsened, whereas the necessary details are preserved for the small ring. The running times for these computations (for a single threaded application on a Core2 Duo E6700 CPU) are shown in Table 5.1. Filtered floating point arithmetic has been used.

The observed runtimes are practical for moderately large data sets, but naturally cannot compete with mesh reconstruction methods that do not come with a topological guarantee (see e.g. [53]) or with medial axis algorithms which are not



(a) Without pruning: 85237 balls

(b) Transparent

(c) Mesh on top of 85237 vertices

(d) After moderate pruning: 4198 balls

(e) Transparent

(f) Mesh on top of 4198 vertices

(g) After heavy pruning: 549 balls

(h) Transparent

(i) Mesh on top of 549 vertices

Figure 5.17: Double torus reconstruction

Figure	5.17abc	5.17def	5.17ghi
Surface balls	55s	55s	55s
Pruning	-	35s	159s
# Remaining balls	85237	4198	549
Weighted α-shape	217s	7s	1s

Table 5.1: Runtimes for the double torus model in Figure 5.17

scalable [72]. Still, our approach compares well with mesh reconstruction methods with guarantee; see e.g. [28]. The strength of our method lies in combining topological correctness with scalability.

Bibliography

[1] Udo Adamy, Joachim Giesen, and Matthias John. Surface reconstruction using umbrella filters. *Computational Geometry: Theory and Applications*, 21(1-2):63–86, 2002.

[2] Pankaj K. Agarwal and Sandeep Sen. Randomized algorithms for geometric optimization problems. In J. Pardalos, S. Rajasekaran, J. Reif, and J. Rolim, editors, *Handbook of Randomized Computation, Kluwer Academic Press, The Netherlands, 2001.* 2001.

[3] Pankaj K. Agarwal and Subhash Suri. Surface approximation and geometric partitions. *SIAM Journal on Computing*, 27(4):1016–1035, 1998.

[4] O. Aichholzer, F. Aurenhammer, T. Hackl, and B. Kornberger. Scalable piecewise linear approximations of 3d medial axes. *Manuscript*, 2009.

[5] Oswin Aichholzer, Franz Aurenhammer, T. Hackl, B. Kornberger, M. Peternell, and H. Pottmann. Approximating boundary-triangulated objects with balls. In *Proceedings of the 23nd European Workshop on Computational Geometry*, volume 23, pages 130–133, 2007.

[6] Oswin Aichholzer, Franz Aurenhammer, Bernhard Kornberger, Simon Plantinga, Günter Rote, Astrid Sturm, and Gert Vegter. Recovering structure from r-sampled objects. In *Eurographics Symposium on Geometry Processing*, 2009.

[7] Nina Amenta and Marshall W. Bern. Surface reconstruction by voronoi filtering. *Discrete & Computational Geometry*, 22(4):481–504, 1999.

[8] Nina Amenta, Sunghee Choi, Tamal K. Dey, and N. Leekha. A simple algorithm for homeomorphic surface reconstruction. *Int. J. Comput. Geometry Appl*, 12(1-2):125–141, 2002.

[9] Nina Amenta, Sunghee Choi, and Ravi Krishna Kolluri. The power crust, unions of balls, and the medial axis transform. *Computational Geometry: Theory and Applications*, 19(2-3):127–153, 2001.

[10] Nina Amenta and Ravi Krishna Kolluri. Accurate and efficient unions of balls. In *Proc. 16th Ann. Symposium on Computational Geometry*, pages 119–128, 2000.

[11] Nina Amenta and Ravi Krishna Kolluri. The medial axis of a union of balls. *Computational Geometry: Theory and Applications*, 20(1-2):25–37, 2001.

[12] Dominique Attali and Jean-Daniel Boissonnat. Complexity of the Delaunay triangulation of points on polyhedral surfaces. *Discrete & Computational Geometry*, 30:437–452, 2003.

[13] Dominique Attali, Jean-Daniel Boissonnat, and Herbert Edelsbrunner. Stability and computation of medial axes—a state-of-the-art report. In T. Müller, B. Hamann, and B. Russell, editors, *Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration*, Springer Series on Mathematics and Visualization, 2007.

[14] Jean-Daniel Boissonnat and Frédéric Cazals. Smooth surface reconstruction via natural neighbour interpolation of distance functions. In *Proc. 16th Ann. Symposium on Computational Geometry*, pages 223–232. 2000.

[15] Jean-Daniel Boissonnat and Frédéric Cazals. Natural neighbor oordinates for points on a surface. *Computational Geometry: Theory and Applications*, 19:155–173, 2001.

[16] Kent B. Bolton. Biarc curves. *Computer-Aided Design*, 7(2):89–92, 1975.

[17] Gareth Bradshaw and Carol O'Sullivan. Adaptive medial-axis approximation for sphere-tree construction. *ACM Transactions on Graphics*, 23(1):1–26, 2004.

[18] Glen E. Bredon. *Topology and geometry*, volume 139 of *Graduate texts in Mathematics*. Springer, 1993.

[19] Kevin Buchin, Simon Plantinga, Günter Rote, Astrid Sturm, and Gert Vegter. Convex approximation by spherical patches. In *Proceedings of the 23nd European Workshop on Computational Geometry*, pages 26–29, Graz, Austria, March 2007.

[20] Wai-San Chan and Francis Y. L. Chin. Approximation of polygonal curves with minimum number of line segments or minimum error. *Int. J. Comput. Geometry Appl*, 6(1):59–77, 1996.

[21] Frdric Chazal and Gert Vegter. Computation of the medial axis of smooth curves with topological guarantees. Technical Report ACS-TR-122102-01, May 2006.

[22] Frédéric Chazal, David Cohen-Steiner, and André Lieutier. Compact sets in euclidean spaces. *Discrete & Computational Geometry*, to appear.

[23] Frédéric Chazal and André Lieutier. Smooth manifold reconstruction from noisy and non-uniform approximation with guarantees. *Computational Geometry: Theory and Applications*, 40(2):156–170, 2008.

[24] G. Das and D. Joseph. The complexity of minimum convex nested polyhedra. In *Proc. 2nd Canad. Conf. Comput. Geom.*, pages 296–301, 1990.

[25] Gautam Das and Michael T. Goodrich. On the complexity of optimization problems for 3-dimensional convex polyhedra and decision trees. *Computational Geometry: Theory and Applications*, 8:123–137, 1997.

[26] Gautam Das and Deborah Joseph. Minimum vertex hulls for polyhedral domains. *Theor. Comput. Sci.*, 103(1):107–135, 1992.

[27] Tamal K. Dey. Curve and surface reconstruction. In Jacob E. Goodman and Joseph O'Rourke, editors, *Handbook of Discrete and Computational Geometry, CRC Press, 1997, 2004*, volume 2. 2004.

[28] Tamal K. Dey, Joachim Giesen, and James Hudson. Delaunay based shape reconstruction from large data. In *Proc. IEEE Symp. Parallel and Large-Data Visualization and Graphics*, pages 19–27, 2001.

[29] Tamal K. Dey and Samrat Goswami. Provable surface reconstruction from noisy samples. *Computational Geometry: Theory and Applications*, 35(1-2):124–141, 2006.

[30] Tamal K. Dey and Wulue Zhao. Approximating the medial axis from the voronoi diagram with a convergence guarantee. *Algorithmica*, 38(1):179–200, 2003.

[31] David H. Douglas and Thomas K. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *The Canadian Cartographer*, 10(2):112–122, December 1973.

[32] R.L. Drysdale. *Generalized Voronoi Diagrams and Geometric Searching*. PhD thesis, Dept. of Computer Science, Stanford University, Stanford, CA., 1979. STAN-CS-79-705.

[33] Scot Drysdale, Günter Rote, and Astrid Sturm. Approximation of an open polygonal curve with a minimum number of circular arcs. In *Proceedings of the 22nd European Workshop on Computational Geometry*, pages 25–28, 2006.

[34] Scot Drysdale, Günter Rote, and Astrid Sturm. Approximation of an open polygonal curve with a minimum number of circular arcs and biarcs. *Computational Geometry: Theory and Applications (Special Issue on the 22nd EuroCG)*, 41:31–47, October 2008.

[35] Herbert Edelsbrunner. The union of balls and its dual shape. *Discrete & Computational Geometry*, 13:415–440, 1995.

[36] Herbert Edelsbrunner. *Geometry and Topology for Mesh Generation*. Cambridge University Press, Cambridge, UK, 2001.

[37] Herbert Edelsbrunner and Ernst P. Mücke. Three-dimensional alpha shape. *ACM Transactions on Graphics*, 13(1):43–72, 1994.

[38] Johannes Eibl. Approximation of planar curves within an asymmetric tolerance band. Master's thesis, Universität Salzburg, Institut für Computerwissenschaften, 2002.

[39] David Eu and Godfried T. Toussaint. On approximating polygonal curves in 2 and 3 dimensions. *Computer Vision, Graphics and Image Processing*, 56(3):231–246, May 1994.

[40] Joachim Giesen and Matthias John. Surface reconstruction based on a dynamical system. *Comput. Graph. Forum*, 21(3), 2002.

[41] Joachim Giesen, Balint Miklos, and Mark Pauly. Medial axis approximation of planar shapes from union of balls: A simpler and more robust algorithm. In Prosenjit Bose, editor, *Proc. Canad. Conf. Comput. Geom.*, pages 105–108. Carleton University, Ottawa, Canada, 2007.

[42] Michael Godau. On the difficulty of embedding planar graphs with inaccuracies. In Roberto Tamassia and Ioannis G. Tollis, editors, *Graph Drawing*, volume 894 of *Lecture Notes in Computer Science*, pages 254–261. Springer, 1994.

[43] L. J. Guibas and S. Y. Oudot. Reconstruction using witness complexes. *Discrete & Computational Geometry*, 40(3):325–356, 2008.

[44] Leonidas J. Guibas, John E. Hershberger, Joseph S. B. Mitchell, and Jack S. Snoeyink. Approximating polygons and subdivisions with minimum link paths. *Intl. J. Comput. Geometry Appl.*, 3(4):383–415, December 1993.

[45] Paul S. Heckbert and Michael Garland. Optimal triangulation and quadric-based surface simplification. *Computational Geometry: Theory and Applications*, 14(1-3):49–65, 1999.

[46] Martin Heimlich and Martin Held. Biarc approximation, simplification and smoothing of polygonal curves by means of voronoi-based tolerance bands. *Int. J. Comput. Geometry Appl*, 18(3):221–250, 2008.

[47] Martin Held and Johannes Eibl. Biarc approximation of polygons within asymmetric tolerance bands. *Computer-Aided Design*, 37(4):357–371, 2005.

[48] Hugues Hoppe, Tony DeRose, Tom Duchamp, John Alan McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. In James J. Thomas, editor, *SIGGRAPH*, pages 71–78. ACM, 1992.

[49] Josef Hoschek and Helmut Pottmann. Interpolation and approximation with developable B-spline surfaces. In Morten Dæhlen, Tom Lyche, and Larry L. Schumaker, editors, *Proceedings of the first Conference on Mathematical Methods for Curves and Surfaces (MMCS-94)*, pages 255–264, Nashville, USA, June 16–21 1995. Vanderbilt University Press.

[50] Philip M. Hubbard. Approximating polyhedra with spheres for time-critical collision detection. *ACM Transactions on Graphics*, 15(3):179–210, 1996.

[51] Hiroshi Imai and Masao Iri. Computational-geometric methods for polygonal approximations of a curve. *Computer Vision, Graphics, and Image Processing*, 36(1):31–41, 1986.

[52] Hiroshi Imai and Masao Iri. Polygonal approximations of a curve – formulations and algorithms. In G. T. Toussaint, editor, *Computational Morphology*, pages 71–86. Elsevier Science Publishers B. V., New York, N.Y., 1988.

[53] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Eurographis Symposium on Geometry Processing*, pages 61–70, 2006.

[54] Hyunjong Ki, Jeongho Shin, Junghoon Jung, Seongwon Lee, and Joon Ki Paik. Surface smoothing for enhancement of 3d data using curvature-based adaptive regularization. In *IWCIA*, pages 488–501, 2004.

[55] Nico Kruithof and Gert Vegter. Envelope surfaces. In *Proc. 22nd Ann. Symposium on Computational Geometry*, pages 411–420, 2006.

[56] Charles Loop and Jim Blinn. Real-time GPU rendering of piecewise algebraic surfaces. *ACM Transactions on Graphics*, 25(3):664–670, 2006.

[57] Mario A. Lopez and Shlomo Reisner. Linear time approximation of 3d convex polytopes. *Computational Geometry: Theory and Applications*, 23(3):291–301, 2002.

[58] Dereck S. Meek and Desmond J. Walton. Approximation of discrete data by G_1 arc splines. *Computer-Aided Design*, 24(6):301–306, 1992.

[59] Dereck S. Meek and Desmond J. Walton. Approximating quadratic NURBS curves by arc splines. *Computer-Aided Design*, 25(6):371–376, 1993.

[60] Dereck S. Meek and Desmond J. Walton. Approximating smooth planar curves by arc splines. *Journal of Computational and Applied Mathematics*, 59:221–231, 1995.

[61] Avraham A. Melkman and Joseph O'Rourke. On polygonal chain approximation. In G. T. Toussaint, editor, *Computational Morphology*, pages 87–95. Elsevier Science Publishers B. V., 1988.

[62] Joseph S. B. Mitchell and Subhash Suri. Separation and approximation of polyhedral objects. *Computational Geometry: Theory and Applications*, 5:95–114, 1995.

[63] Partha Niyogi, Stephen Smale, and Shmuel Weinberger. Finding the homology of submanifolds with high confidence from random samples. *Discrete & Computational Geometry*, 39:419–441, 2008.

[64] Joseph O'Rourke. *Computational Geometry in C*. Cambridge University Press, Cambridge, UK, 1994.

[65] Mark Pauly, Niloy J. Mitra, Joachim Giesen, Markus H. Gross, and Leonidas J. Guibas. Example-based 3D scan completion. In *Proc. 21st Ann. Symposium on Computational Geometry*, pages 23–32, 2005.

[66] Les A. Piegl. Curve fitting algorithm for rough cutting. *Computer-Aided Design*, 18(2):79–82, March 1986.

[67] Vishwa Ranjan and Alain Fournier. Matching and interpolation of shapes using unions of circles. *Computer Graphics Forum*, 15(3):129–142, 1996.

[68] Jens Schönherr. Smooth biarc curves. *Computer-Aided Design*, 25(6):365–370, 1993.

[69] Andrei Sharf and Ariel Shamir. Feature-sensitive 3D shape matching. In *Computer Graphics International*, pages 596–599. IEEE Computer Society, 2004.

[70] B. Q. Su and Da-You Liu. *Computational Geometry — Curve and Surface Modeling*. Academic Press, Boston, 1989.

[71] Jacob Subag and Gershon Elber. Piecewise developable surface approximation of general NURBS surfaces, with global error bounds. In Myung-Soo

Kim and Kenji Shimada, editors, *GMP*, volume 4077 of *Lecture Notes in Computer Science*, pages 143–156. Springer, 2006.

[72] Avneesh Sud, Mark Foskey, and Dinesh Manocha. Homotopy-preserving medial axis simplification. *Int. J. Comput. Geometry Appl.*, 17:423–451, 2007.

[73] Godfried T. Toussaint. On the complexity of approximating polygonal curves in the plane. In *Proceedings IASTED, International Symposium on Robotics and Automation, Lugano, Switzerland*, 1985.

[74] Craig A. Tovey. A simplified NP-complete satisfiability problem. *Discrete Appl. Math.*, 8:85–89, 1984.

[75] Kasturi R. Varadarajan. Approximating monotone polygonal curves using the uniform metric. In *Proc. 12th Ann. Symposium on Computational Geometry*, pages 311–318, 1996.

[76] Yuandong Yang, Oliver Brock, and Robert N. Moll. Efficient and robust computation of an approximated medial axis. In *Proc. 9th ACM Symposium on Solid Modeling and Applications*, pages 15–24. ACM, 2004.

[77] Millan K. Yeung and Desmond J. Walton. Curve fitting with arc splines for NC toolpath generation. *Computer-Aided Design*, 26(11):845–849, 1994.

[78] Jun-Hai Yong, Shi-Min Hu, and Jia-Guang Sun. A note on approximation of discrete data by G1 arc splines. *Computer-Aided Design*, 31(14):911–915, 1999.

[79] Jun-Hai Yong, Shi-Min Hu, and Jia-Guang Sun. A note on approximation of discrete data by G1 arc splines. *Computer-Aided Design*, 31(14):911–915, 1999.

[80] Uri Zwick. All pairs shortest paths using bridging sets and rectangular matrix multiplication. *J. ACM*, 49(3):289–317, 2002.

List of Figures

Zusammenfassung

Die Approximation eines geometrischen Objektes hat zum Ziel, ein komplexes Objekt durch ein vereinfachtes zu ersetzen, ohne die Charakteristika des ursprünglichen Objektes zu verlieren. In der Dissertation werden Approximationsalgorithmen für polygonale Kurven im 2-dimensionalen Raum und für Flächen im 3-dimensionalen Raum vorgestellt. Im ersten Teil behandeln wir die Approximation von polygonalen Kurven. Die Approximation einer polygonalen Kurve kann durch eine ebenfalls polygonale, aber nun einfachere, Kurve (eine Kurve mit weniger Segmenten), oder durch eine Kurve höherer Ordnung erfolgen. Man verbindet typischer Weise zwei Optimierungsprobleme mit der Approximation von polygonalen Kurven. Man möchte entweder die Anzahl der verwendeten Teilstücke der approximierenden Kurve zu einer gegebenen Fehlertoleranz minimieren oder man versucht den Approximationsfehler der approximierenden Kurve zu minimieren, hierbei wird die Anzahl der zu verwendeten Teilstücke vorgegeben. Während zahlreiche Algorithmen in der Literatur bekannt sind, die diese Optimierungsprobleme für die Approximation mit polygonalen Kurven lösen, so stellen sich die gleichen Fragen auch für die Approximation mit Kurven höheren Grades. Wir stellen in dieser Arbeit sowohl einen Algorithmus vor, der eine polygonale Kurve mit einer minimalen Anzahl and Kreisbögen, als auch einen Algorithmus der eine polygonale Kurve mit der minimalen Anzahl Biarcs (zwei glatt miteinander verbundene Kreisbögen) approximiert. Im zweiten Teil der Arbeit wenden wir uns der Approximation von Flächen im 3-dimensionalen Raum zu. Wir stellen zwei Algorithmen vor, die aus einer gegebenen Punktmenge die Oberfläche eines Objektes rekonstruieren. Zuerst betrachten wir nur Punktemengen in konvexer Lage und zeigen, dass die Approximation dieser Punktmengen mit einer minimalen Anzahl an Kugelkappen NP-schwer ist. Da dieses Problem vermutlich nicht optimal zu lösen ist, präsentieren wir einen inkrementellen Greedy-Algorithmus, der eine gegebene Punktemenge in konvexer Lage mit einer gekrümmten Fläche aproximiert. Als letztes betrachten wir die Rekonstruktion einer Fläche aus einer Punktmenge in nicht konvexer Lage. Hierbei nehmen wir die Idee des inkrementellen Ansatzes auf. Um eine Startkonfiguration für einen inkrementellen Algorithmus zu haben, stellen wir einen Algorithmus vor, der aus einer Punktprobe einer nicht konvexen Fläche ein möglichst kleines (aber nicht garantiert das kleinste) Polytop erzeugt, das folgende Eigenschaften hat: die Mittelachse der ursprünglichen Fläche ist im Polytop enthalten, das Polytop verläuft durch eine

Teilmenge der Probenpunkte und die ursprüngliche Topologie bleibt erhalten. Wie zeigen wie dieser Algorithmus nicht nur für die Erzeugung eines Startpolytops genutzt werden kann, sondern auch um skalierbare Rekonstruktionen der Fläche zu erzeugen.

Lebenslauf

Der Lebenslauf ist in der Online-Version aus Grnden des Datenschutzes nicht enthalten.