Fast Distance Computation Using Quadratically Supported Surfaces

Margot Rabl and Bert Jüttler

Abstract We use the class of surfaces with quadratic polynomial support functions in order to define bounding geometric primitives for shortest distance computation. The common normals of two such surfaces can be computed by solving a single polynomial equation of degree six. Based on this observation, we formulate an algorithm for computing the shortest distance between enclosures of two moving or static objects by surfaces of this type. It is demonstrated that the performance of this algorithm compares favourably with methods for computing the distance between two ellipsoids, which can also be used as bounding primitives for distance computation and collision detection.

1 Introduction

Collision detection and shortest distance computation between static or dynamic geometric objects is one of the fundamental problems in robotics and computer graphics, especially for applications such as path planning, computer games, simulation of machining processes and animation, see e.g. [4, 5, 8, 9, 11]. In order to formulate efficient algorithms, the objects are often replaced by (hierarchies of) geometric primitives, such as axis-aligned or oriented bounding boxes (AABBs or OBBs), or k-DOPS. The computational effort for collision detection and distance computation depends on the number of these primitives, on the computationally effort for analyzing a pair of two primitives, and on the costs for adapting these primitives to dynamic objects (which may require a frequent re-computation, e.g. in the case of AABBs for a moving body).

The majority of the existing methods relies on geometric primitives with piecewise linear boundaries, since then the elementary operations of interference detec-

Margot Rabl and Bert Jüttler
Institute of Applied Geometry, Johannes Kepler University, A-4040 Linz, Austria
e-mail: {margot.rabl,bert.juettler}@jku.at

tion and distance computation between them become particularly simple. These simple primitives, however, do not represent free-form objects very well, and alternative approaches have been explored. On the one hand, one may use hybrid techniques which combine bounding primitives with the exact representations [3]. On the other hand, it seems to be promising to explore other bounding primitives with non-linear boundaries.

In particular, the use of ellipsoids (and more general algebraic surfaces) has been proposed in the literature [1, 2, 7, 10]. Methods for generating collections of bounding ellipsoids for 3D objects have been discussed in [12]. As observed there, for certain types of objects (in particular with free-form boundaries), the use of bounding ellipsoids leads to a smaller number of bounding primitives compared to OBBs. However, the distance computation between two ellipsoids leads to an algebraic problem of high degree [10] - it can eventually be formulated as the problem of finding all roots of a polynomial of degree 24.

We present a new class of bounding geometric primitives for distance computation which performs better than ellipsoids. As a first advantage, these primitives can represent both elliptic (locally convex) and hyperbolic regions of free-form surfaces. As another advantage, the computational costs of computing the shortest distance between two of our primitives is much smaller than in the case of ellipsoids; it can be formulated as a root finding problem for a polynomial of degree 6. This fact has been established recently [14]. In the present paper we use this observation in order to formulate an algorithm for shortest distance computation between enclosures of objects by the new geometric primitives. We demonstrate its performance by several examples and compare it with the best available techniques for distance computation between ellipsoids.

2 Enclosing Moving Objects by QSS

We recall the notion of quadratically supported surfaces (QSS) and discuss the computation of common normals of two such surfaces.

We consider the function which is obtained by restricting a quadratic polynomial $f:\mathbb{R}^3\to\mathbb{R}$ to the unit sphere $\mathbb{S}^2$,

$$f:\ \mathbb{S}^2\to\mathbb{R}:\ \mathbf{n}\mapsto f(\mathbf{n})=\mathbf{n}^\top\mathbf{A}\mathbf{n}+\mathbf{a}^\top\mathbf{n}, \tag{1}$$

where $\mathbf{n}=(x,y,z)^\top$ denotes the unit normal, with coefficients $a_1,\dots,a_9\in\mathbb{R}$,

$$\mathbf{A}=\frac{1}{2}\begin{pmatrix}2a_1 & a_4 & a_5\\ a_4 & 2a_2 & a_6\\ a_5 & a_6 & 2a_3\end{pmatrix}\ \text{and}\ \mathbf{a}=\begin{pmatrix}a_7\\ a_8\\ a_9\end{pmatrix} \tag{2}$$

Since we are only interested in the restriction to the unit sphere, we can assume that the constant term in (1) has been incorporated into the quadratic one, as a multiple of $\mathbf{n}^\top\mathbf{n}=1$.

The envelope of the two-parameter family of planes with unit normals $\mathbf{n} \in \mathbb{S}^2$ and support distances $f(\mathbf{n})$ (distances to the origin) defines a quadratically supported surface (QSS), see [15, 16]. It can be parameterized as

$$\begin{aligned} \mathbf{x}_f : \mathbb{S}^2 &\to \mathbb{R}^3 : \\ \mathbf{n} &\mapsto \mathbf{x}_f(\mathbf{n}) = f(\mathbf{n})\mathbf{n} + \nabla f(\mathbf{n}) - \left(\nabla f(\mathbf{n})^\top \mathbf{n}\right)\mathbf{n} \\ &\qquad\qquad = -(\mathbf{n}^\top \mathbf{A}\mathbf{n})\,\mathbf{n} + 2\mathbf{A}\mathbf{n} + \mathbf{a}, \end{aligned} \tag{3}$$

where the domain is the unit sphere $\mathbb{S}^2$. More precisely, we obtain an *oriented surface*, where each point $\mathbf{x}_f$ has an associated unit normal vector $\mathbf{n}$.

The class of QSS is closed under translations, rotations, scaling, offsetting, and orientation reversal. For instance, a rigid body displacement

$$\delta : \mathbb{R}^3 \to \mathbb{R}^3 : \mathbf{x} \mapsto \mathbf{v} + U\mathbf{x} \tag{4}$$

with rotation matrix $\mathbf{U} \in \mathrm{SO}(3)$ and translation vector $\mathbf{v}$ transforms the support function (1) into

$$f'(\mathbf{n}) = \mathbf{n}^\top \mathbf{U}\mathbf{A}\mathbf{U}^\top \mathbf{n} + (\mathbf{U}\mathbf{a} + \mathbf{v})^\top \mathbf{n}. \tag{5}$$

Moving QSS have been analyzed in [13], where it was shown that they support an exact computation of the characteristic curves on envelope surfaces.

On the other hand, the reversal of the orientation of the normals modifies the support function according to

$$f''(\mathbf{n}) = -f(-\mathbf{n}) = -\mathbf{n}^\top \mathbf{A}\mathbf{n} + \mathbf{a}^\top \mathbf{n}. \tag{6}$$

By combining a suitable rotation, translation and a scaling, the support of a QSS can be transformed into the normal form

$$f(\mathbf{n}) = x^2 + k_1 y^2 + k_2 z^2 \tag{7}$$

with coefficients $k_1, k_2 \in \mathbb{R}$. Hence, any QSS can be identified with a point in the (k_1, k_2)-plane. For instance, surfaces of revolution occur if $k_1 = k_2$, $k_1 = 1$ or $k_2 = 1$.

In order to identify QSS without self-intersections, we analyzed their singularities and obtained the following results. If the QSS defined by the support function f has no self-intersections, then it is the boundary surface of a simply connected subset $F \subset \mathbb{R}^3$. There exist two different types of simply connected QSS (the proof will be presented in the doctoral thesis of the first author):

- If the shape parameters (k_1, k_2) lie in the interior of the hexagon bounded by the six lines

  $$k_2 = \frac{1}{2}, \quad k_2 = \frac{1}{2}k_1, \quad k_1 = 2, \quad k_2 = 2, \quad k_2 = 2k_1 \quad \text{and} \quad k_1 = \frac{1}{2}, \tag{8}$$

  then the bounded volume of the associated QSS is convex and its boundary possesses no singularities.
- If the shape parameters satisfy either

$$\frac{1}{2} < k_1 < 2 \wedge k_2 < 0 \text{ or } \frac{1}{2} < k_2 < 2 \wedge k_1 < 0 \text{ or } k_1 > 2k_2 \wedge k_2 > 2k_1, \quad (9)$$

then the volume which is enclosed by the QSS is non-convex. However, the boundary of the volume has singular curves, and the orientation of the unit normals $\mathbf{n}$ along the boundary may change along these singularities[1].

Fig. 1 visualizes the four regions (8), (9) in the (k_1, k_2)-plane, where simply connected QSS are supported. Furthermore four examples of QSS corresponding to the values indicated by circles are shown.

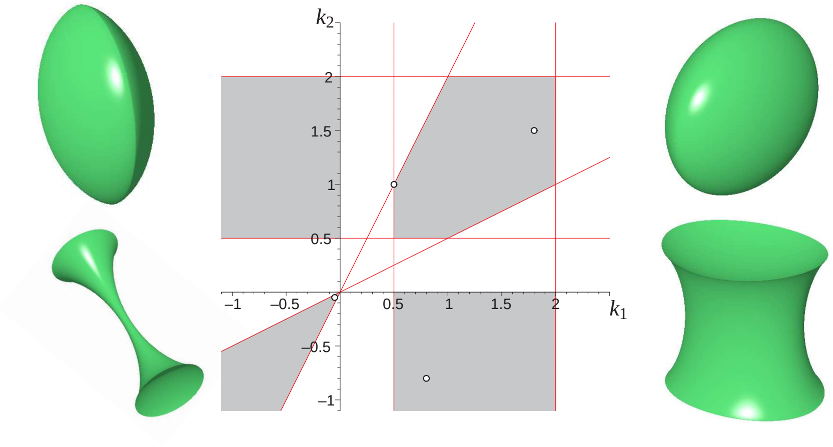

Fig. 1 Center: Parameter domain (k_1, k_2) of a support function in normal form. The grey domains indicate shapes without self-intersections. Left and right: Simply connected QSS for different values of (k_1, k_2) (indicated by the small circles in the innermost figure). The singularities are located on one circle, on two circles and along two non-planar curves (in counterclockwise order from top left). The top right QSS is convex and free of singularities.

Now we consider two QSS with the support functions f (see (1)) and

$$g(\mathbf{n}) = \mathbf{n}^\top \mathbf{B}\mathbf{n} + \mathbf{b}^\top \mathbf{n} \quad (10)$$

with coefficients $b_1, \ldots, b_9 \in \mathbb{R}$. Two points $\mathbf{x}_f(\mathbf{n}_0)$ and $\mathbf{x}_g(\mathbf{n}_0)$ of the surfaces are said to have a common normal, if their difference vector has the same direction as $\mathbf{n}_0$, i.e., if the vectors

$$\mathbf{x}_f(\mathbf{n}_0) - \mathbf{x}_g(\mathbf{n}_0) \quad \text{and} \quad \mathbf{n}_0 \quad (11)$$

are linearly dependent. Using (3) and the corresponding equation for g, this condition leads to a system of three quadratic equations for $\mathbf{n}_0$,

$$[2(\mathbf{A} - \mathbf{B})\mathbf{n}_0 + \mathbf{a} - \mathbf{b}] \times \mathbf{n}_0 = \mathbf{0} \quad (12)$$

[1] The normals may point inside of the object on one side of a singular curve, and outside on the other side.

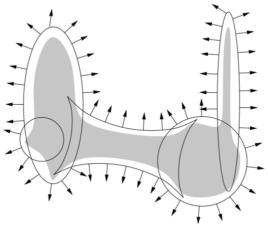

Fig. 2 A valid enclosure with outward-pointing normals (two-dimensional sketch).

which have to be satisfied by a unit vector $\mathbf{n}_0$. As shown in [14], the three quadric surfaces (12) (where $\mathbf{n}_0 = (x,y,z)^\top$) intersect in a rational cubic curve, which can be parameterized easily. The unit normals that correspond to a point-pair with common normals are then found by intersecting this rational cubic curve with the unit sphere, leading to a polynomial equation of degree 6. Consequently, *two QSS possess at most 6 point-pairs with common normals*, and these point-pairs can be computed by solving a polynomial equation of degree 6. This compares favourably with (e.g.) two quadric surfaces, where the same problem leads to a polynomial of degree 24 [10].

3 Valid enclosures by QSS and distance computation

In order to compute the approximate distance between two objects, we want to cover them by a collection of QSS, such that the shortest distance of the two enclosures is realized at a common normal of a pair of QSS. We describe the assumptions in the following definition.

Definition 1. A covering of an object $\mathscr{O}$ by a collection $\mathscr{F} = \bigcup_{i=1}^{k} F_i$ of k simply connected QSS F_i with support functions f_i is called a *valid enclosure with outward-* (resp. *inward-*) *pointing normals* if

1. the object is contained in the interior, $\mathscr{O} \subseteq \mathscr{F}$,
2. all singularities of the bounding QSS are contained in the interior of $\mathscr{F}$ and
3. the outermost boundary has outward- (resp. inward-) pointing normals.

These conditions ensure, that the smallest distance of two valid enclosures can be found by checking all common normals.

Lemma 1. *Consider two disjoint objects which are covered by valid enclosures* $\mathscr{F} = \bigcup_{i=1}^{k} F_i$ *and* $\mathscr{G} = \bigcup_{j=1}^{l} G_j$ *of QSS, which have inward-pointing and outward-pointing normals, respectively. Then the minimum distance D satisfies*

$$D \geq \min_{\substack{i=1,\dots,k \\ j=1,\dots,l}} d_{i,j} \tag{13}$$

where d_{ij} *is the minimum distance of any point-pair on two QSS* F_i *and* G_j *with common normals. Moreover, if the two objects are equal to their enclosures* $\mathscr{F}$ *and* $\mathscr{G}$*, then the inequality becomes an equation.*

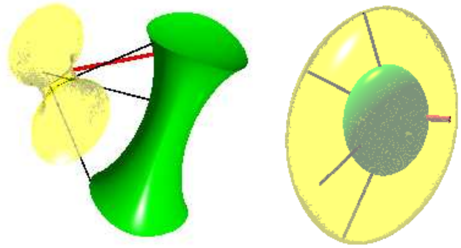

Fig. 3 Common normals and shortest distance (red) between two static simply connected QSS. Left: two separate non-convex QSS with 4 common normals. Right: Two nested convex QSS with 6 common normals.

Proof. It suffices to observe that all edges of the outermost boundary of the valid enclosures are concave edges. The minimum distance between the two enclosures is never realized at one of these edges. Consequently, it occurs at a point-pair with common normals between two QSS F_i and G_j. □

Lemma 1 suggests the following algorithm for computing the shortest distances between two valid enclosures $\mathscr{F}$ and $\mathscr{G}$.

```
Input: two valid enclosures with support functions $f_i$ and $g_j$.
1. For all pairs $(i, j)$:
    Find all common normals $\mathbf{n}_0$.
    For all common normals $\mathbf{n}_0$:
        Evaluate the distances between the points $\mathbf{x}_{f_i}(\mathbf{n}_0)$ and $\mathbf{x}_{g_j}(\mathbf{n}_0)$.
    Compute the minimum $d_{ij}$ of these distances.
2. Compute the minimum of all $d_{ij}$.
Output: the minimum distance and the corresponding point-pair.
```

We implemented this algorithm in C++, where we used the Jenkins-Traub algorithm [6] for computing the roots of the polynomial equation of degree 6. The roots are computed with machine precision. So far, we do not exploit the possible temporal or spatial coherence. This could be done by using other root finding methods, such as a Newton-type solver.

The performance of our algorithm is demonstrated by two examples.

Example 1: Common normals of two static QSS

First we consider only two QSS, which do not necessarily form valid enclosures of disjoint objects (this would require both of them to be convex.) Figure 3 presents two examples with 4 and 6 common normals, respectively. The computation times are reported in Table 1.

Table 1 Running times for distance computation between two QSS.

number of common normals	2	4	6
computation time on a 2 GHz PC	$20-22\mu s$	$29-31\mu s$	$38-40\mu s$

For our method, the computation time is currently dominated by the Jenkins-Traub algorithm for root finding. The time needed for setting up the cubic equation is negligible. This also explains the large variations of the computing time, which depends on the number of common normals. Typically, the computation time is around $30\mu s$. So far, the case of six common normals has only been observed for nested QSS which are not relevant for distance computation. We compare our results to those obtained by [1], for computing the distance between two quadric surfaces. They use a Newton-type algorithm to find the common normals and need $56.25\mu s$, composed of $40\mu s$ preprocessing and $16.25\mu s$ "main algorithm" on a 1.7 GHz PC, which corresponds to approximately $47.81\mu s$ on our hardware. Quadric surfaces have the same number of geometric degrees of freedom (i.e. 9) as QSS, but the shortest distance computation is much slower, even when compared with our non-optimized method.

Example 2: Common normals and shortest distances between a moving enclosure and an obstacle.

We consider a moving object covered by two convex and one non-convex QSS that is subject to a rigid body motion. For each position, we compute a lower bound of the distance between the moving object and a static obstacle, which is bounded by a collection of three convex QSS (Fig. 4).

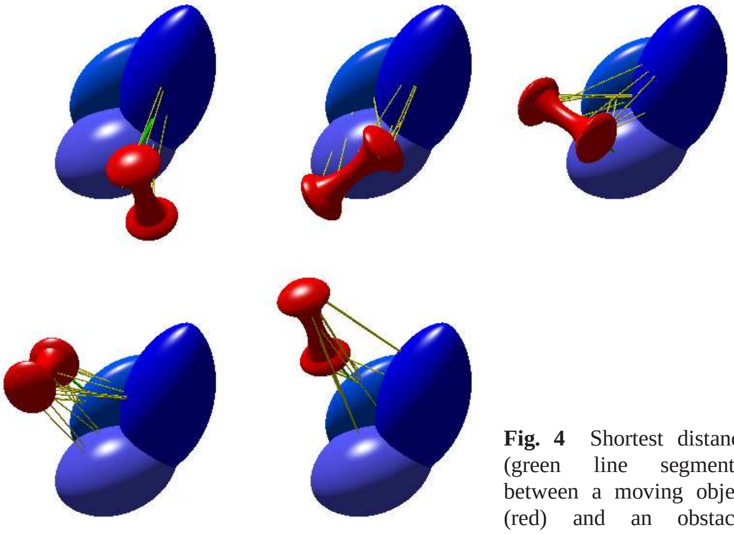

Fig. 4 Shortest distance (green line segments) between a moving object (red) and an obstacle (blue), both enclosed by a collection of QSS.

For each time step of the motion, we have to compute 9 times all common normals of a pair of QSS. We applied the algorithm to $10,000$ time steps, where the computation took $2.75s$ on a 2 GHz PC. This corresponds to an average computation time of about $30.5\mu s$ per QSS-pair and time step. The time needed for transforming the QSS bounding the moving object (two 3×3 matrix multiplications and one vector addition) is negligible.

Clearly, the application of our method requires a (preferably automatic) method for creating valid enclosures of 3D objects by QSS, similar to the method in [12] for ellipsoids. This may be a topic for future research.

Acknowledgment. Supported by the Austrian Science Fund, project S9202.

References

1. Chen X-D, Yong J-H, Zheng G-Q, Paul J-C, Sun J-G (2006) Computing minimum distance between two implicit algebraic surfaces. Computer-Aided Design 38: 1053–1061.
2. Choi Y-K, Wang W, Kim M-S (2003) Exact Collision Detection of Two Moving Ellipsoids under Rational Motions. In: Proc. Int. Conf. Robotics and Automation, IEEE Press, 349–354.
3. Chou W, Xiao J (2006) Real-time and Accurate Multiple Contact Detection between General Curved Objects. In: Proc. Int. Conf. on Intelligent Robots and Systems, IEEE Press, 556–561.
4. Cohen J, Lin M, Manocha D, Ponamgi M (1995) I-COLLIDE: An interactive and exact collision detection system for large-scale environments. In: Proc. ACM Interactive 3D Graphics Conference. ACM Press, 189–196.
5. Eberly DH (2001) Game Engine Design. Academic Press.
6. Jenkins MA (1975) Algorithm 493 - Zeros of a Real Polynomial [C2], ACM Trans. Math. Software 1: 178–189.
7. Ju M, Liu J, Shiang S, Chien Y (2001) A Novel Collision Detection Method Based on Enclosed Ellipsoid. In: Proc. Int. Conf. Robotics and Automation, IEEE Press, 2897–2902.
8. Klosowski J, Held M, Mitchell JSB, Sowizral H, Zikan K (1998) Efficient collision detection using bounding volume hierarchies of k-DOPs. IEEE Transactions on Visualization and Computer Graphics 4: 21–36.
9. Latombe J-C (1991) Robot Motion Planning. Kluwer Academic Publishers, Boston.
10. Lennerz, C and Schömer, E (2002) Efficient Distance Computation for Quadratic Curves and Surfaces. In: Proc. Geometric Modeling and Processing, IEEE Press, 60–69.
11. Lin MC, Gottschalk S (1998) Collision detection between geometric models: A survey. In: Cripps R (Ed.), Proc. IMA Conf. on Mathematics of Surfaces, Information Geometers: 37–56.
12. Lu L, Choi YK, Wang W and Kim M-S (2007) Variational 3D Shape Segmentation for Bounding Volume Computation, Computer Graphics Forum 26: 329–338.
13. Oberneder M, Jüttler B, Gonzalez-Vega L (2008) Exact envelope computation for moving surfaces with quadratic support functions. In Lenarčič J, Wenger P, Advances in Robot Kinematics - Analysis and Design, Springer, 283–290.
14. Rabl M, Gonzalez-Vega L, Jüttler B, Schröcker H-P (2009) Oriented Bounding Surfaces with at most Six Common Normals. In: Proc. Int. Conf. Robotics and Automation, to appear. Available as FSP report no. 77 at http://www.industrial-geometry.at/techrep.php
15. Sabin M (1974) A Class of Surfaces Closed under Five Important Geometric Operations. Technical Report VTO/MS/207, British Aircraft Corporation. Available at http://www.damtp.cam. ac.uk/user/na/people/Malcolm/vtoms/vtos.html.
16. Šír Z, Gravesen J, Jüttler B (2008) Curves and surfaces represented by polynomial support functions. Theoretical Computer Science 392: 141–157.