Sweeping an Arrangement of Quadrics in 3D *

Bernard Mourrain Jean-Pierre Técourt Monique Teillaud

Abstract

1 Introduction

Arrangements are the underlying structure of many applications, especially in robot motion planning. They have been extensively studied in the literature (see [Hal97] for a survey).

The arrangement of a set of objects $\mathcal{S}$ in $\mathbb{R}^d$ is the decomposition of $\mathbb{R}^d$ into cells of dimensions $0, 1, \ldots, d$ induced by $\mathcal{S}$. The topology of an arrangement is often quite complex, and the description of a given cell can be of non-constant size. Therefore, vertical decompositions are often used, allowing to partition the space into simpler constant sized cells (for a complete bibliography, we refer to [SH02]). A sweep-based algorithm was followed in [dBGH96, SH02] to produce a vertical decomposition of an arrangements of triangles in $\mathbb{R}^3$.

The manipulation of algebraic surfaces plays an important role in solid modeling. Geismann *et al.* presented two methods to compute a given cell in an arrangement of quadrics [GHS01]. The first method uses projection techniques based on resultants, while the second method uses solid modeling techniques.

We propose here a sweeping algorithm to compute effectively the arrangement of a set of quadrics in $\mathbb{R}^3$.

2 Overview

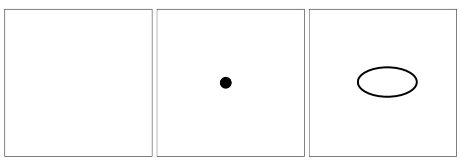
Let $\mathcal{S} = \{Q_i, i = 1, \ldots, n\}$ be a set of n quadrics. We denote by Q_i both a quadric and its equation. Let ∇Q_i be the gradient vector of Q_i. We assume that no quadric is a product of planes.

We choose a generic direction (say z) and we sweep $\mathcal{S}$ by a plane in this direction. Every z-section of the arrangement is an arrangement of conics in the plane. We initialize the sweep at some chosen value of z. Let us consider the different types of events where the topology of the z-section is changing during the sweep.

a) $Q_{i_1} = 0, \partial_x(Q_{i_1}) = 0, \partial_y(Q_{i_1}) = 0$: horizontal tangent plane.

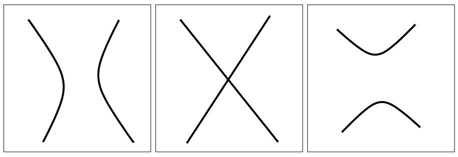
More precisely, depending on the signature of Q_{i_1}, three events can appear:

i. (3,1), (1,3):

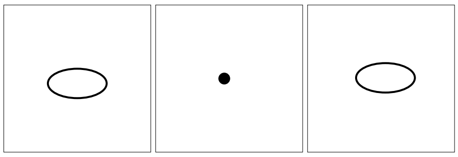


*GALAAD, INRIA, BP 93, 06902 Sophia Antipolis cedex, FRANCE
{Bernard.Mourrain, Jean-Pierre.Tecourt, Monique.Teillaud}@sophia.inria.fr
http://www-sop.inria.fr/galaad/
This work is partially supported by the IST Programme of the EU as a Shared-cost RTD (FET Open) Project under Contract No IST-2000-26473 (ECG - Effective Computational Geometry for Curves and Surfaces)
http://www-sop.inria.fr/prisme/ECG/

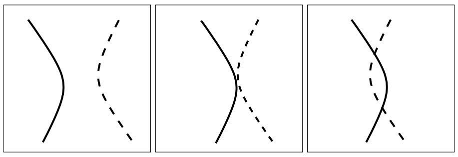
ii. (2,2):



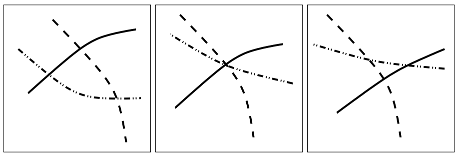
iii. (2,1), (1,2):



b) $Q_{i_1} = 0, Q_{i_2} = 0, (\nabla Q_{i_1} \bigwedge \nabla Q_{i_2})_z = 0$: horizontal tangent for the intersection curve between Q_{i_1} and Q_{i_2}.



c) $Q_{i_1} = 0, Q_{i_2} = 0, Q_{i_3} = 0$: intersection point of three quadrics.

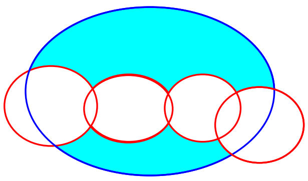


We do not consider degenerate cases such as intersections of more than three quadrics at the same point.

3 From cells to "trapezoids"

The first idea consists in characterizing each 2-dimensional cell of the arrangement of conics in a section by sign conditions. For one or two conics, the sign conditions are determined by the equations of the quadrics and the equations of lines depending on the quadrics. See [MTT02] for more details.

In the case when cells are defined by more than 3 quadrics, the following picture shows that two different cells (the two gray cells) can be characterized by exactly the same sign conditions.



To solve this issue, we choose to compute a "trapezoidal" decomposition of the arrangement in the z-section, as explained in the following paragraph.

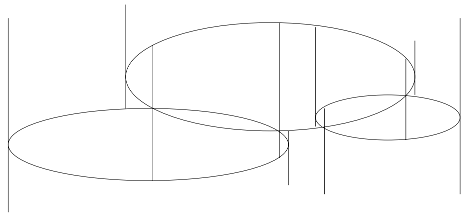
Trapezoids. We draw segments parallel to the y-axis. This is done in a very similar way as done usually for the trapezoidal map in the case of a planar arrangement of line segments. A vertical segment will be drawn through:

- intersection points between two conics

- points where the tangent to the conic is parallel to the y-axis.

We obtain "trapezoids" of constant size description: the boundary of each trapezoid consists into two vertical walls, a ceiling and a floor. Both the ceiling and the floor are conic arcs.

Deciding whether a point lies in a trapezoid, reduces to compare the x-coordinates of the point and the walls and then for a fixed x, to compute the sign of the conics defining the ceiling and the floor or the sign of rational expressions formed on their coefficients.



The drawback is that maintaining the vertical decomposition introduces additional events that have no meaning in the 3D arrangement, but the big advantage of this decomposition is that all events described in Section 2, except events of type (a.i), can be easily detected during the algorithm: each time a new trapezoid is created, we compute the z for which it disappears. All the events of Section 2 are some of these events.

When a trapezoid disappears, the 2D arrangement needs to be updated: the trapezoid is replaced by other trapezoids, and its neighbors are modified, too. Enumerating the different types of trapezoids is quite easy, as well as the way they need to be updated, depending on the type of event that cause them to disappear. Details are omitted in this abstract.

Only events of type (a.i) will be precomputed and sorted. When such an event is encountered, a point location has to be performed. The decomposition into trapezoids allows to locate such a point easily in practice, either in a naive way by testing all the trapezoids, or by walking along a line.

3D decomposition. Another advantage of the trapezoidal decomposition is that it induces a decomposition of the arrangement of quadrics in $\mathbb{R}^3$ into simple regions, that are the regions swept by the trapezoids. The decomposition we get with our method is not quite the same as the so-called *vertical decomposition* [SA95].

We skip the discussion on the combinatorial complexity in this abstract. The data structures used are roughly similar to the ones described in [SH02]. We chose to focus on algebraic aspects.

4 Algebraic aspects

The events of type (a.i) in Section 2 are precomputed by solving algebraic equations of degree 2, and they are sorted.

Location in the trapezoidal map. As written above, deciding whether a point lies in a trapezoid reduces to compute signs of rational expressions in the coefficients of the quadrics, and to compare the x-coordinates of the point and the vertical walls. So, the point location for an event of type (a.i) performs such evaluations of signs at points whose coordinates belong to an algebraic extension of degree at most 2, and comparisons of degree 2 and 4 algebraic numbers.

Detecting and comparing new events. A trapezoid is defined by two vertical walls, a floor and a ceiling. To predict how a trapezoid will disappear, we need to compute when its floor and its ceiling collide, or when its vertical walls coincide.

The worst case, in terms of algebraic degree, is achieved by the events when the x-coordinate of the intersection between two conics coincides with the x-coordinate of the intersection between two other conics. This leads to the computation of points of intersection of 4 quadrics in a space of dimension 4, whose coordinates lie in an algebraic extension of degree at most 16. The coordinates of the intersection points are rational functions of these algebraic numbers.

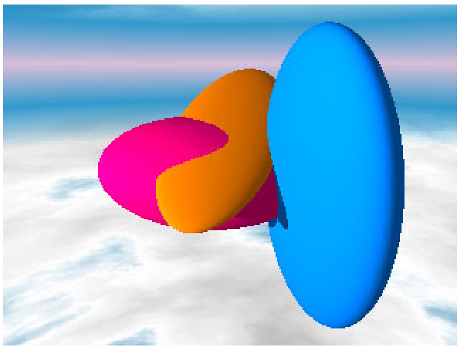
In order to sort the events according to the z-direction, we have to determine the sign of the difference of two algebraic numbers. In the worst case, we are interested in algebraic numbers of degree 16 belonging to independent algebraic extensions of the initial field. So, the difference is in an algebraic extension of degree 256.

Preliminary experimental results. These computations can be done in practice with the SYNAPS library[1]. Let us consider the arrangement of the following 3 quadrics:

$$272x^2 + 96xy + 192xz + 32y^2 + 64yx2 + 64z^2 - 571.2x - 142.4y - 252.8z + 323.64 = 0$$
$$128x^2 + 1152y^2 - 1024yz + 256z^2 - 144x - 886.4y + 358.4z + 220.12 = 0$$
$$64x^2 + 256y^2 + 128z^2 - 64x - 288y - 160z + 143 = 0$$

We have considered the events corresponding to a change in the topology of the cross section (See Section 2). The events corresponding to changes in the trapezoidal map are not computed in these preliminary tests. An approximation of the events (a), (b), (c) is computed with the following running times on a PC workstation (i686, 2.2 GHz, 256 M):
(a) 3×2 real solutions (0.01s).
(b) $3 \times 8 = 24$ complex solutions and 6 are real (0.06s).
(c) 8 complex solutions and 2 real (0.02s).
Then, the events are sorted according to the z-coordinate as follows:

```
(a) [0.825000,0.700000,0.287500]
(a) [0.562500,0.544649,0.359835]
(a) [0.500000,0.562500,0.448223]
(b) [0.498552,0.561349,0.448234]
(b) [0.687835,0.570199,0.508852]
(b) [0.677133,0.617014,0.519616]
(c) [0.676862,0.612181,0.521687]
(c) [0.638126,0.657542,0.685372]
(b) [0.534420,0.666721,0.719519]
(b) [0.662072,0.686211,0.723158]
(b) [0.627783,0.558545,0.776837]
(a) [0.500000,0.562500,0.801777]
(a) [0.562500,0.780351,0.890165]
(a) [0.675000,0.300000,0.912500]
```



References

[dBGH96] M. de Berg, Leonidas J. Guibas, and D. Halperin. Vertical decompositions for triangles in 3-space. *Discrete Comput. Geom.*, 15:35–61, 1996.

[GHS01] N. Geismann, M. Hemmer, and E. Schömer. Computing a 3-dimensional cell in an arrangement of quadrics: Exactly and actually! In *Proc. 17th Annu. ACM Sympos. Comput. Geom.*, pages 264–273, 2001.

[Hal97] D. Halperin. Arrangements. In Jacob E. Goodman and Joseph O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 21, pages 389–412. CRC Press LLC, Boca Raton, FL, 1997.

[MTT02] Bernard Mourrain, Jean-Pierre Tecourt, and Monique Teillaud. Algebraic methods for dealing with 3d implicit quadrics. Technical Report ECG-TR-182105-02, INRIA Sophia-Antipolis, 2002.

[SA95] Micha Sharir and P. K. Agarwal. *Davenport-Schinzel Sequences and Their Geometric Applications.* Cambridge University Press, New York, 1995.

[SH02] H. Shaul and D. Halperin. Improved construction of vertical decompositions of three-dimensional arrangements. In *Proc. 18th Annu. ACM Sympos. Comput. Geom.*, pages 283–292, 2002.

[1] `http://www-sop.inria.fr/galaad/logiciels/synaps/`