# The Convex Hull of Ellipsoids


Nicola Geismann
Fachrichtung 6.2 Informatik
Universität des Saarlandes
66123 Saarbrücken, Germany
nicola@cs.uni-sb.de

Michael Hemmer Elmar Schömer
Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85
66123 Saarbrücken, Germany
hemmer | schoemer@mpi-sb.mpg.de



## Abstract

The treatment of curved algebraic surfaces becomes more and more the focus of attention in Computational Geometry. We present a video that illustrates the computation of the convex hull of a set of ellipsoids. The underlying algorithm is an application of our work on determining a cell in a 3-dimensional arrangement of quadrics, see [3]. In the video, the main emphasis is on a simple and comprehensible visualization of the geometric aspects of the algorithm. In addition, we give some insights into the underlying mathematical problems.


## The Algorithm

There are well known efficient and robust algorithms for the calculation of the convex hull of a set of points or a set of spheres ([1], [2]). In our video, we show how to compute the convex hull of a set of ellipsoids with exact arithmetic (Fig. 1, 2). The video consists of three different parts. First, we illustrate how the problem of computing the convex hull can be reduced to the problem of calculating a cell in a 3-dimensional arrangement of quadrics. Next, a further reduction to planar arrangements and their treatment is visualized. Finally, we show the connection between the topology of the convex hull in the primal space and the topology of the computed cell in the dual space.

## Dualization

The problem of calculating the convex hull of a set of ellipsoids can be reduced via duality to the problem of computing a cell in an arrangement of ellipsoids, paraboloids, and hyperboloids, quadrics for short (Fig. 3). The quadrics partition affine space in a natural way into four different types of maximal connected regions: *cells* are on either side of each quadric, *faces* lie on exactly one quadric, *edges* are on the intersection curve of two quadrics and *vertices* are intersection points of three or more quadrics. We have to compute the boundary of the cell that lies in the interior of each quadric, i.e. the boundary of the *intersection cell* (Fig. 4). For further information about the dualization of quadrics see [4]. The connection between the convex hull in the primal space and the intersection cell in the dual space is very well visualizable in two dimensions. This is what is shown in the first part of the video (Fig. 5).



*SCG'01,* June 3-5, 2001, Medford, Massachusetts, USA.
ACM 1-58113-357-X/01/0006.

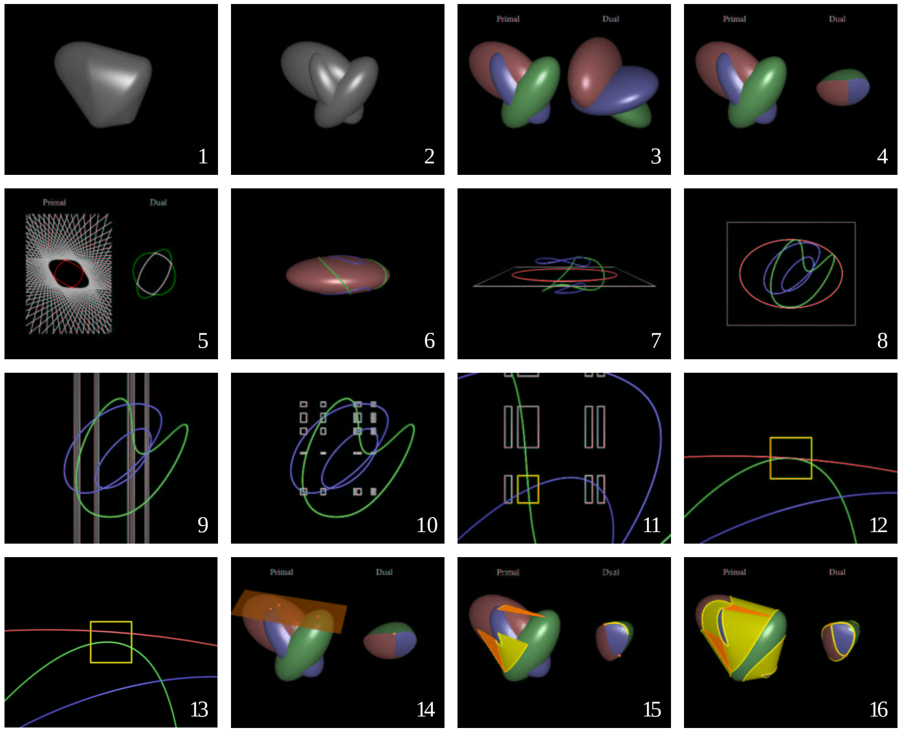

## Computing the Arrangement

In the next part the video shows how to compute the topological description of the intersection cell of a set of quadrics exactly. The video illustrates the main procedure of the algorithm. For the mathematical details we refer the reader to [3]. We have implemented our algorithm. The following pictures of the planar arrangements arise from the visualization print-out during the execution of the program. We realized the project using the basic data types of LEDA [6] and the rational polynomial class as well as the resultant computation of MAPC [5].

In order to compute the intersection cell, we determine for each quadric a complete description of the arrangement restricted to its surface (Fig. 6). It is easy to see that these separate topological descriptions can be assembled to a unified one of the intersection cell with little combinatorial and administrative effort. The problem is that in general there is no rational parameterization of the intersection curve of two quadrics. Therefore we project the boundary of the quadric and all its intersection curves with other quadrics into the $(x, y)$-plane (Fig. 7). This can be done using an algebraic tool called *resultant* and yields a set of planar algebraic curves of degree at most 4 (Fig. 8). During the projection we loose the spatial information but with the help of ray shooting we can recover it afterwards.

Now we have to compute a topological description of the planar arrangement. The most serious problem we face is that usually the intersection points of two curves have irrational coordinates. That is why we cannot deal with them directly. Instead, we apply a resultant computation to each pair of curves obtaining a univariate polynomial in $x$. With a Sturm sequence computation we isolate the real roots of the univariate polynomial. The rational intervals bounding the real roots define small stripes with rational $x$-coordinates, parallel to the $y$-axis that cover all intersection points (Fig. 9). We do the same for the $y$-coordinates. The intersection of the stripes yields boxes with rational corners (Fig. 10).

Then we test whether an intersection point of the two curves takes place inside a box or not. To decide this, we only have at our disposal the discrete information of what happens at the boundary of the box. This information may be sufficient. For example, if there is no hit from any curve, we know that there cannot be an intersection point inside. Or, if there are two hits with each curve and these hits alternate, we know that there must be an intersection point inside the box (Fig. 11). But this method, called *box-hit*




*counting*, is not always suitable: If we consider tangential intersection points of two curves or self intersection points of one curve, the scenario on the boundary of the box cannot tell us anything about an intersection point inside the box (Fig. 12, 13). We have developed a method to overcome these difficulties by locally introducing a new curve to the arrangement in the box and then again using box-hit counting. This method will only fail in very few cases in which we can apply a global criterion.

## Correspondence between Primal and Dual Space

In the last part of our video we show how the intersection cell in the dual space corresponds to the convex hull in the primal space. A point in the dual space dualizes to a plane in the primal space. What about a vertex (also called triple intersection point in the video) in the dual space? Its dual is a plane that touches each of the three involved ellipsoids in exactly one point (Fig. 14). The three points define a triangle and the area lying on the tangential plane bounded by the triangle will be part of the convex hull. Analogously, each point on an edge of the intersection cell defines a line connecting the two involved ellipsoids (Fig. 15). All lines together form a developable surface. So the convex hull consists of three different types of surfaces: parts of the original quadric surfaces, triangles which arise from the dual vertices, and developable surfaces which arise from the dual edges (Fig. 16).

## Acknowledgments

We would like to thank Christian Lennerz and Jan Kautz for their technical support and Susan Hert for revising the script.

## 1. REFERENCES

[1] J.-D. Boissonnat, A. Cérézo, O. Devillers, J. Duquesne, and M. Yvinec. An algorithm for constructing the convex hull of a set of spheres in dimension *d*. *Comput. Geom. Theory Appl.*, 6:123–130, 1996.

[2] http://www.cgal.org.

[3] N. Geismann, M. Hemmer, and E. Schömer. Computing a 3-dimensional cell in an arrangement of quadrics: Exactly and actually! *SOCG*, 2001.

[4] C.-K. Hung and D. Ierardi. Constructing convex hulls of quadratic surface patches. In *Proc. 7th Canad. Conf. Comput. Geom.*, pages 255–260, 1995.

[5] J. Keyser, T. Culver, D. Manocha, and S. Krishnan. MAPC: A library for efficient and exact manipulation of algebraic points and curves. In *Proc. 15th Annu. ACM Sympos. Comput. Geom.*, pages 360–369, 1999.

[6] K. Mehlhorn and S. Näher. *LEDA – A Platform for Combinatorial and Geometric Computing*. Cambridge University Press, 1999.