The Predicates for the Voronoi Diagram of Ellipses[*]

Ioannis Z. Emiris[†] Elias P. Tsigaridas[†] George M. Tzoumas[†‡]
emiris@di.uoa.gr et@di.uoa.gr geotz@di.uoa.gr

Department of Informatics and Telecommunications
National and Kapodistrian University of Athens, Greece

Abstract

This paper examines the computation of the Voronoi diagram of a set of ellipses in the Euclidean plane. We propose the first complete algorithms, under the exact computation paradigm, for the predicates of an incremental algorithm: κ_1 decides which one of 2 given ellipses is closest to a given exterior point; κ_2 decides the position of a query ellipse relative to an external bitangent line of 2 given ellipses; κ_3 decides the position of a query ellipse relative to a Voronoi circle of 3 given ellipses; κ_4 determines the type of conflict between a Voronoi edge, defined by 4 given ellipses, and a query ellipse. The paper is restricted to non-intersecting ellipses, but the extension to arbitrary ones is possible.

The ellipses are input in parametric representation or constructively in terms of their axes, center and rotation. For κ_1 and κ_2 we derive optimal algebraic conditions, solve them exactly and provide efficient implementations in C++. For κ_3 we compute a tight bound on the number of complex tritangent circles and use the parametric form of the ellipses in order to design an exact subdivision-based algorithm, which is implemented on Maple. This approach essentially answers κ_4 as well. We conclude with current work on optimizing κ_3 and implementing it in C++.

1 Introduction

Computational geometry for curved objects relies on predicates implemented by algebraic operations. In this paper we study the Voronoi diagram of ellipses in the exact computation paradigm. The distance of an exterior point to an ellipse is defined to be the minimum Euclidean distance to any point of the ellipse. We design and implement exact and complete algorithms for the predicates needed in the framework of *abstract* Voronoi diagrams [22] and, more particularly, the incremental algorithm in [20]. To be more precise, the algorithm computes the *Delaunay graph*, since no computation of Voronoi vertices or edges is necessary. Still, if one wishes to draw the diagram with fixed precision, the algorithm and our methods provides sufficient information.

Our final goal is CGAL[1] software for constructing the Voronoi diagram of ellipses, based on the CGAL implementation for circles [12], which uses the same incremental algorithm. Hence the crucial question is to analyze and implement the predicates for ellipses. Some of the presented predicates are also needed in computing the visibility complex and the convex hull of ellipses.

Voronoi diagrams have been studied extensively, however the bulk of the existing work in the plane concerns point or linear sites. One relevant work computes the diagram of convex polygons [23], with an approach similar to ours, since the algorithm "moves" on the objects' boundary using pruning techniques. Recent efforts have extended Voronoi diagrams to the case where the sites are curves (e.g. [1, 3]) or have non-empty interior [8]. In particular, the diagram of circles has been implemented in CGAL [12]; see also [4, 21]. Anton [3] examines κ_3 for the diagram of ellipses but his algebraic system's mixed volume is too large, hence leading to high complexity. His matrix methods for solving the system seem slower than ours and do not guarantee exactness.

In [19], an optimal combinatorial algorithm is derived for constructing Voronoi diagrams of strictly convex rounded sites in $\mathbb{R}^3$, but the predicates are not considered. In [7], a dynamic algorithm is described for constructing the power diagrams of points in $\mathbb{R}^d$. This specializes to the diagram of circles or spheres, but does not seem to cover ellipses. Another line of work, which has been quite successful, is to approximate the curved sites by polygons, e.g. [5]. In [6] the authors compute a polygonal approximation of a Voronoi diagram at different levels of detail. We expect that applications, such as navigation among objects, shall benefit from an exact diagram of ellipses, given the ability of ellipses to model, quite accurately, different kinds of obstacles.

Perhaps the work coming closest to ours is [18].

[*] SCG'06, June 5–7, 2006, Sedona, Arizona, USA

[†] All authors acknowledge partial support by IST Programme of the EU as a Shared-cost RTD (FET Open) Project under Contract No IST-006413-2 (ACS - Algorithms for Complex Shapes) and by PYTHAGORAS, project 70/3/7392 under the EPEAEK program of the Greek Ministry of Educational Affairs and EU.

[‡] Partially supported by State Scholarship Foundation of Greece, Grant No. 4631.

[1] www.cgal.org

The authors essentially trace the bisectors in order to compute the Voronoi cells of arbitrary curves up to machine precision. Their algorithm uses floating point arithmetic; they claim that their software works well in practice. Although they argue that their algorithm can be extended to exact arithmetic, they do not explain how. For instance, they do not discuss degenerate configurations. Our implementations are exact but can also run with any prescribed precision.

A second motivation comes from visibility problems among ellipses [17], or pairwise disjoint bounded convex sets of constant complexity [2]. In particular, the computation and characterization (as external or internal) of all bitangents of two ellipses is of independent interest. Our approach obtains this characterization and hence also answers κ_2. The additional information is used by subsequent calls to κ_2.

A sketch of our work on the Voronoi circle of 3 ellipses and κ_3 appeared in [15]. Presently, we offer a full investigation of the problem dealing with both degenerate and non-degenerate configurations. We study the case of *non-intersecting* ellipses, which is straightforward to generalize to arbitrary ellipses and even pseudo-circles [20]. We assume that the input ellipses are given *parametrically*, or *constructively* in terms of their axes, center and rotation, all being rational (see next section). This permits to switch to an implicit representation.

The 4 predicates of the incremental algorithm in [20] are:

- (κ_1) given two ellipses and a point outside of both, decide which is the ellipse closest to the point,
- (κ_2) given 2 ellipses, decide the position of a third one relative to a specific external bitangent of the first two,
- (κ_3) given 3 ellipses, decide the position of a fourth one relative to the (external tritangent) Voronoi circle of the first 3; this is the InCircle predicate,
- (κ_4) given 4 ellipses, compute the part of the bisector that changes due to the insertion of a 5th ellipse.

Our first contribution are algorithms for κ_1 and κ_2 that are optimal in terms of algebraic degree. In fact, for κ_2, we compute and characterize all bitangents of two ellipses which provides additional information. Both algorithms are exact, complete and implemented in C++. The tangency points are or are not computed depending on whether we deal with the parametric or the implicit representation.

Using the implicit representation, we obtain the first tight bound on the number of *complex* tritangent circles to 3 ellipses, namely 184. The number of real tritangent circles remains open. However, this approach did not lead to an efficient algorithm. Hence we turned to parametric representation. The Voronoi circle is specified by the intersection of bisectors, at any desired accuracy. This is achieved by refining the interval expressing the 3 tangency points until the predicate can be decided; in fact, all tangency points are expressed as a function of one of them. Exactness is guaranteed by root separation bounds from the equations of the implicit representation of the problem.

We present and implement in Maple a customized subdivision-based algorithm for κ_3, which "moves" on the border of parametrically defined ellipses. This exploits the underlying geometry and avoids computing the Voronoi circle. Hence, our code is faster than applying generic state-of-the-art software to approximate the Voronoi circle. Our dedicated solver contains univariate root isolation and comparison, as well as sign evaluation of polynomials in up to 2 variables.

The subdivision-based algorithm essentially answers κ_4, as well. This is the first complete solution of how to implement the Voronoi diagram of ellipses (via the Delaunay graph) in the exact computation paradigm.

The paper is organized as follows. The next section discusses representation issues. In sec. 3 we give algorithms to decide predicates κ_1 and κ_2. Sec. 4 studies the Voronoi circle from the implicit representation viewpoint. The parametric representation is considered in sec. 5 in order to yield an algorithm for κ_3. Predicate κ_4 is settled in sec. 6. Sec. 7 illustrates our implementations with various tests and the last section concludes with future work.

2 Representation

An ellipse has the following implicit equation:

$$E(x,y) := ax^2 + 2bxy + cy^2 + 2dx + 2ey + f \in \mathbb{Q}[x,y] \tag{1}$$

Let the length of the major and minor axes be $2\alpha, 2\beta$, respectively. Let (x_c, y_c) be its (rational) center. Throughout this paper, we use the following parametric representation:

$$\begin{aligned} x(t) &= x_c + \alpha\left(\tfrac{1-w^2}{1+w^2}\right)\left(\tfrac{1-t^2}{1+t^2}\right) - \beta\left(\tfrac{2w}{1+w^2}\right)\left(\tfrac{2t}{1+t^2}\right) \\ &= x_c + \tfrac{-\alpha(1-w^2)t^2 - 4\beta wt + \alpha(1-w^2)}{(1+w^2)(1+t^2)} \\ y(t) &= y_c + \alpha\left(\tfrac{2w}{1+w^2}\right)\left(\tfrac{1-t^2}{1+t^2}\right) + \beta\left(\tfrac{1-w^2}{1+w^2}\right)\left(\tfrac{2t}{1+t^2}\right) \\ &= y_c + 2\tfrac{-\alpha wt^2 + \beta(1-w^2)t + \alpha w}{(1+w^2)(1+t^2)}, \end{aligned} \tag{2}$$

where $t = \tan(\theta/2) \in (-\infty, \infty)$, θ is the angle that traces the ellipse, $w = \tan(\omega/2)$, and ω is the rotation angle between the major and horizontal axes. This representation leaves out of the boundary a single point, called the *i*-point.

The symmetric ellipse (with respect to its center) is $\bar{x}(t) = -x(-t) + 2x_c$ and $\bar{y}(t) = -y(t) + 2y_c$. We call it the *twin* ellipse. Every point of an ellipse is different from its twin point, including the *i*-point. We

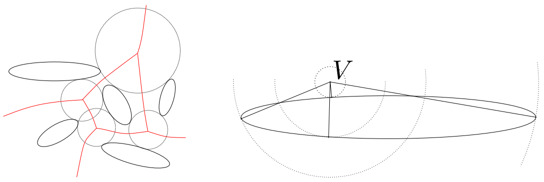


Figure 1: Left: Voronoi diagram of 5 ellipses. Right: an example of a point with 4 normals.

denote by $E_t(\alpha, \beta, w, x_c, y_c)$ or simply E_t, an ellipse parameterized by t and by $\bar{E}_t$ its twin ellipse. The coefficients of (1) are polynomials in the coefficients of (2):

$$\begin{array}{rcl}
\chi & = & y_c w^2 + 2x_c w - y_c \\
\psi & = & x_c w^2 - 2y_c w - x_c \\
(1+w^2)^2 a & = & 4w^2\alpha^2 + (w-1)^2(w+1)^2\beta^2 \\
(1+w^2)^2 b & = & 2(\alpha-\beta)(\alpha+\beta)w(w-1)(w+1) \\
(1+w^2)^2 c & = & 4w^2\beta^2 + (w-1)^2(w+1)^2\alpha^2 \\
(1+w^2)^2 d & = & -2w\chi\alpha^2 - (w-1)(w+1)\psi\beta^2 \\
(1+w^2)^2 e & = & +2w\psi\beta^2 - (w-1)(w+1)\chi\alpha^2 \\
(1+w^2)^2 f & = & \chi^2\alpha^2 + \psi^2\beta^2 - (1+w^2)^2\alpha^2\beta^2
\end{array} \tag{3}$$

Note that χ, ψ express the equations of the major and minor axes evaluated at (x_c, y_c). The following quantities are *invariant* under rotation and translation: $J_1 = a + c = \alpha^2 + \beta^2$, $J_2 = ac - b^2 = \alpha^2\beta^2$, while $J_4 = J_2(x_c^2 + y_c^2 - J_1)$ is invariant under rotation. Now, $x_c = (be - dc)/J_2$, $y_c = (bd - ae)/J_2$.

Since the ellipse is given in parametric form, or constructively (rational axes, center and w), the above equations transform it to an implicit form.

3 Predicates κ_1 and κ_2

For κ_1, we are given 2 ellipses and a point outside of both, and we wish to find the one closest to the point, under the Euclidean metric. To lower bound the inherent algebraic complexity, take a point V outside an ellipse; it may have up to 4 normals to the ellipse, depending on its position relative to the *evolute* curve (which is a stretched astroid). There are 4, 3 or 2 normals if V lies inside the evolute, on the evolute but not at a cusp or, respectively, V is a cusp or outside the evolute.

Consider an ellipse E, represented algebraically, and point $V = (v_1, v_2)$ outside E. We denote by $C(V, \sqrt{s})$ a *circle* centered at V with radius equal to $\sqrt{s}, s > 0$. We express the *Euclidean distance* $\delta(V, E)$ between V and E by the *smallest* positive value of $\sqrt{s}$ for which C is *tangent* to E. In comparing distances, it is sufficient to consider the squared distance s.

Let us express a conic as $[x, y, 1]M[x, y, 1]^T$, for an appropriate matrix M. Then E, C correspond to

$$A = \begin{pmatrix} a & b & d \\ b & c & e \\ d & e & f \end{pmatrix}, \qquad B(s) = \begin{pmatrix} 1 & 0 & -v_1 \\ 0 & 1 & -v_2 \\ -v_1 & -v_2 & v_1^2 + v_2^2 - s \end{pmatrix}.$$

Their *pencil* is $\lambda A + B$, and their characteristic polynomial is

$$\varphi(\lambda) = \det(\lambda A + B(s)) = J_2^2\lambda^3 + c_2(s)\lambda^2 + c_1(s)\lambda + s, \tag{4}$$

where $c_2(s) = J_2 s - T(v_1, v_2)$, $c_1(s) = J_1 s - E(v_1, v_2)$ and $T(v_1, v_2) = J_2[(v_1 - x_c)^2 + (v_2 - y_c)^2 - J_1]$. Using E and T for $E(v_1, v_2)$ and $T(v_1, v_2)$, its discriminant is:

$$\begin{array}{rcl}
\Delta(s) & = & J_2^2(J_1^2 - 4J_2)\, s^4 + \\
& & 2J_2(9J_1J_2^2 - J_1^2 T + 6J_2 T - 2J_1^3 J_2 - J_1 J_2 E)\, s^3 + \\
& & (-18J_2^3 E + 4J_1 J_2 E T - 27J_2^4 + J_1^2 T^2 - 18J_1 J_2^2 T \\
& & + J_2^2 E^2 + 12J_1^2 J_2^2 E - 12J_2 T^2)\, s^2 + \\
& & 2(2T^3 - J_1 E T^2 - 6J_1 J_2^2 E^2 + 9J_2^2 E T - J_2 E^2 T)\, s + \\
& & E^2(T^2 + 4J_2^2 E)
\end{array} \tag{5}$$

A circle is externally tangent to an ellipse iff $\varphi(\lambda)$ has a positive double root [25, thm.8], [16, sec.4]. This is recognized by the vanishing of Δ, which is a univariate polynomial (in s) of degree 4.

Now $\delta(V, E)$ is the square-root of the smallest positive zero of $\Delta(s)$. The degree of the coefficients of $\Delta(s)$, in v_1, v_2 and the parameters of E, is 6, 8, 10, 12, and 14, in order of decreasing power in s. The degree of every coefficient of $\Delta(v_1, v_2, s)$ (when viewed as a trivariate polynomial) in the parameters of E is 6.

Proposition 3.1 *Given ellipses E_1, E_2 and V outside both of them, we decide which ellipse is closest to V by comparing two algebraic numbers of degree 4.*

This degree is optimal with respect to the algebraic numbers compared. Optimality follows since the worst case requires us to work with algebraic numbers of 4th degree. In sec. 7 we report on our exact implementation in C++.

If we restrict ourselves to the parametric representation we need to compare two algebraic numbers of degree 8.

Now we examine κ_2. This predicate decides the position of a query ellipse relative to one bitangent line to 2 given ellipses. In fact, our algorithm provides additional information on the characterization of the bitangent lines, which allows us to answer future predicates.

Consider a (non-vertical) line $L : y = ux + v$ and ellipse E represented in implicit form. Let us substitute y in E; then, for L to be tangent to E, its discriminant Λ must vanish: The bitangent line is expressed by system $\Lambda_1 = \Lambda_2 = 0$, where Λ_i corresponds to the i-th ellipse. The system has ≤ 4 real roots corresponding to the 4 bitangents. A vertical bitangent line is treated in an easier way.

Now we can answer κ_2 as follows. Given an external bitangent $\bar{L}: \; y = \bar{u}x + \bar{v}$, we can determine the relative position of a query ellipse E_3 with respect to this line by computing the discriminant of $\bar{L}$ and E_3, $\Lambda_3(\bar{u}, \bar{v})$. Λ_3 is negative, zero, or positive iff E_3 has 0, 1 or 2 common points with $\bar{L}$ respectively. In the first 2 cases, the sign of $\bar{L}(x, y)$, evaluated at the center of E_3, specifies the side of $\bar{L}$ on which E_3 lies. The predicate is implemented in C++, see timings in sec. 7.

Now, let us consider the tangent at point $(x(t), y(t))$ of the *parametric* ellipse E_t. The implicit equation of this line is $(y - y(t))x'(t) - (x - x(t))y'(t) = 0$. If we replace $x(t)$ and $y(t)$ from (2) we obtain a polynomial of degree 2 with respect to t. We replace x, y with $x(r), y(r)$ from ellipse E_r and obtain a quadratic polynomial with respect to r the solutions of which correspond to the points where the tangent line of E_t intersects E_r. For this line to be tangent to both ellipses, the discriminant $\Lambda_{tr}(t)$ of the polynomial should vanish. A bitangent line is external to both ellipses iff its equation yields the same sign when evaluated at an interior point of each ellipse. For E_t, the sign is always positive, because the equation evaluates to $2\alpha\beta(1 + w^2)(1 + t^2)$. Hence, to determine the type of a bitangent line, it suffices to compute the sign of a quadratic polynomial, evaluated at an algebraic number of degree 4. This degree is optimal with respect to the algebraic numbers employed.

Now let $t_1 < t_2 < t_3 < t_4$ be the solutions of $\Lambda_{tr}(t)$. Let μ correspond to an internal bitangent and ϵ to an external one. Then (t_1, t_2, t_3, t_4) correspond to a cyclic permutation of $(\mu\epsilon\epsilon\mu)$. Given 2 ellipses, in order to determine the permutation of their bitangents, it suffices to determine the type of exactly two bitangents. Hence, we arrive at the following:

Theorem 3.2 *The relative position of E_s with respect to an external bitangent of ellipses E_t, E_r reduces to the sign of $\Lambda_{ts}(t)$, which has degree 4, over $\hat{t}$, which is a root of $\Lambda_{tr}(t)$ (also of degree 4). Now $sign(\Lambda_{ts}(\hat{t})) = -1, 0,$ or 1 iff E_s does not intersect, is tangent to, or intersects the bitangent respectively.*

4 Implicit approach to κ_3

Given 3 ellipses, we consider their external tritangent circle, known as their Voronoi circle. If there are two such circles, we assume that one is specified. We wish to decide the position of a fourth ellipse relative to this circle. This section considers all ellipses in implicit form and applies certain algebraic techniques, discussed in [11].

Let $\sqrt{s}$ be the radius of the tritangent circle and (v_1, v_2) its center. Using the discriminant as above for each of the 3 ellipses, we get

$$\Delta_1(v_1, v_2, s) = \Delta_2(v_1, v_2, s) = \Delta_3(v_1, v_2, s) = 0. \quad (6)$$

Lemma 4.1 *A solution $(\bar{v}_1, \bar{v}_2, \bar{s})$ of system (6) corresponds to an external tritangent circle iff $\bar{s}$ is the smallest positive root of all $\Delta_i(\bar{v}_1, \bar{v}_2, s)$, $i = 1, 2, 3$. If s_0^-, s_0^+ are the smallest and largest positive roots of $\Delta_0(\bar{v}_1, \bar{v}_2, s)$, where Δ_0 corresponds to the query ellipse, then:*

- $\bar{s} \leq s_0^- \Leftrightarrow$ *the query ellipse is outside the circle and is tangent iff $\bar{s} = s_0^-$,*
- $\bar{s} \in (s_0^-, s_0^+) \Leftrightarrow$ *the query ellipse intersects the circle,*
- $\bar{s} \geq s_0^+ \Leftrightarrow$ *the query ellipse is inside the circle and is tangent iff $\bar{s} = s_0^+$.*

Proof. Let $(\bar{v}_1, \bar{v}_2, \bar{s})$ be a solution of the system. ($\Rightarrow$): Let $\bar{s}$ be the smallest positive root of all $\Delta_i(\bar{v}_1, \bar{v}_2, s)$, $i = 1, 2, 3$. Then the circle $(\bar{v}_1, \bar{v}_2, \bar{s})$ is externally tangent to all three ellipses, therefore it is an external tritangent circle. ($\Leftarrow$): Let the circle $(\bar{v}_1, \bar{v}_2, \bar{s})$ be an external tritangent circle to all three ellipses. Since this cirlce is externally tangent to each one of the ellipses, $\bar{s}$ will be the smallest positive root of each $\Delta_i(\bar{v}_1, \bar{v}_2, s)$, $i = 1, 2, 3$.

Now consider all circles $C(\bar{v}_1, \bar{v}_2, s)$, as s grows from zero to infinity, assuming that point $(\bar{v}_1, \bar{v}_2)$ lies outside a query ellipse E_0. When $s = 0$, C is a point outside E_0. When s is at infinity, C becomes an infinite circle enclosing E_0. As s grows from zero, it passes from the roots of $\Delta_0(\bar{v}_1, \bar{v}_2, s)$. At each one of these (at most four) points, C is tangent to E_0. When $s < s_0^-$, C is outside E_0, and when $s > s_0^+$, C encloses E_0. In all other cases, it intersects E_0 due to the topology of the two closed curves C and E_0. □ Among the solutions of this system, the external tritangent circle of interest may or may *not* have the smallest radius.

To reduce the mixed volume of system (6), we remove solutions at infinity by setting

$$q = v_1^2 + v_2^2 - s. \quad (7)$$

Now the discriminant system becomes

$$\Delta_1(v_1, v_2, q) = \Delta_2(v_1, v_2, q) = \Delta_3(v_1, v_2, q) = 0, \quad (8)$$

and has mixed volume 184. We have computed the resultant as a polynomial of degree 184 in q. Adding eq. (7) to this system yields a system in v_1, v_2, q, s with mixed volume 184.

Since each Δ_i is a discriminant, an equivalent system to (8), with the same mixed volume, contains $\varphi_i = 0, \frac{\partial}{\partial \lambda_i}\varphi_i = 0$, for $i = 1, 2, 3$, where φ_i is the characteristic polynomial of ellipse i and the Voronoi circle. This system has substantially smaller coefficients than (8). We solved this system numerically with `PHCpack` [2] which implements homotopy continuation, and found up to 22 real solutions[3]. The total number of complex solutions was 184.

Theorem 4.2 *Three ellipses admit at most 184 complex tritangent circles. This is tight since there are triplets attaining this number.*

Proof. The mixed volume provides an upper bound while the degree of the resultant (of our example) gives a lower bound. □

Recall that in the case of 3 circles, the number of tritangent circles is 8 and the corresponding predicate is of algebraic degree 2. Our theorem generalizes to all types of conics, according to F. Sottile [4]. The interesting open question is how many of these circles can be real. F. Ronga suggests a construction where three conics have at least 136 real tritangent circles. However, we have not been able to achieve such a configuration with three disjoint ellipses.

With bitsize as in table 2, the timings of PHCpack were 23.85, 34.52, 38.36, and 38.44 sec respectively. In [3], the time to solve the system describing the Voronoi circle is several minutes. We also tried the iCOs interval-arithmetic solver [5] on the system of Δ_i's with $B = 60$. It detects a degeneracy for κ_3 (three ellipses and a query one all externally tangent to the same circle) in about 213 sec on a 1GHz P3. Recently, with some preliminary experiments (thanks to D. Lazard) exploiting the Gröbner base computation, we were able to isolate all real roots of the system in about 2 min with GB-Rs.

5 Parametric approach to κ_3

We use the parametric representation of ellipses to study the external bitangent circles. In subsec. 5.1 we shall apply this discussion to external tritangent circles.

We express the Voronoi vertex by the intersection of 2 bisectors. The *bisector* of two ellipses is the locus of points at equal distance from the two ellipses. Given ellipses E_t, E_r and points P, Q on each of them, the *bisector* is obtained as the intersection V of the normal lines at the ellipses, at P, Q, when $|\overrightarrow{PV}| = |\overrightarrow{QV}|$. This expresses all points on the bisector except for a finite number of them, namely where the two normals are parallel.

Point $V(v_1(t,r), v_2(t,r))$ is the solution of a linear system of two equations, expressing the normals respectively at points with parameter values t and r. A point defined by parameter value t will also be referred to as point t, or $t \in E_t$. The coordinates' denominator D_{tr} vanishes iff the normals are parallel to each other.

The bisector is

$$\begin{aligned} B(t,r) &= \big(v_1(t,r) - x(t)\big)^2 + \big(v_2(t,r) - y(t)\big)^2 - \\ &\quad \big(v_1(t,r) - x(r)\big)^2 - \big(v_2(t,r) - y(r)\big)^2 \quad (9) \end{aligned}$$

which is rational in t, r with denominator D_{tr}. In the case of ellipses, the numerator is a bivariate polynomial of degree 6 in t and 6 in r. It can be shown that it also vanishes when D_{tr} vanishes. Therefore it includes both bisector points at infinity as well as points where the normal vectors of the two ellipses coincide (i.e. at the minimum distance between two ellipses). We now consider the bitangent circles.

Proposition 5.1 *Given 2 ellipses and a point on the first, there may exist up to 6 real bitangent circles, tangent at the specific point. This bound is tight.*

Proof. If we fix t, equation (9) has 6 complex solutions with respect to r. Therefore 6 is an upper bound for the number of possible real bitangent circles. Moreover, a configuration of two ellipses that have 6 real bitangent circles can be attained, see fig. 2. □ Note that only one such circle is *external* to both ellipses. We call this unique external bitangent circle the *Apollonius* circle of the 2 ellipses, e.g. the third circle from the right in fig. 2. The Voronoi circle of 3 ellipses is where 3 Apollonius circles coincide.

Given ellipses E_t, E_r as in fig. 4, the tangency points of any Apollonius circle lie inside their Convex Hull (CH). Thus, for the parameterization (2), there is at least an i-point of $E_t, E_r, \bar{E}_t, \bar{E}_r$ that does not lie inside CH. This implies that we can always search for a Voronoi circle within a *continuous* range on the boundary of an ellipse or its twin.

Now, consider all bitangent circles to E_t, E_r, tangent at point t on E_t. Also, consider the lines from t tangent to E_r at points r_1, r_2. They define two arcs on E_r. Arc (r_1, r_2), whose interior points lie on the same side of line $r_1 r_2$ as t, is called a *visible arc.*

Property 5.2 *Visible arc* (r_1, r_2) *contains only tangency points of bitangent circles at* t*, which are externally tangent to* E_r*. These include the Apollonius circle of* E_t, E_r*, tangent at* $t \in E_t$*.*

Proof. From a point Q inside the visible arc (fig. 3), an internally tangent circle to E_r cannot be tangent at t, because the tangent line at Q leaves t and E_r on different hyperplanes. External bitangent circles

[2] http://www.math.uic.edu/∼jan/PHCpack/

[3] We checked the real roots by hand.

[4] Personal communication, 2004.

[5] http://www-sop.inria.fr/coprin/ylebbah/icos/

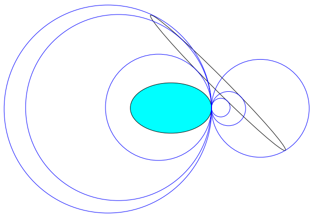


Figure 2: The 6 bitangent circles: The Apollonius circle is the 4^{th} from the left

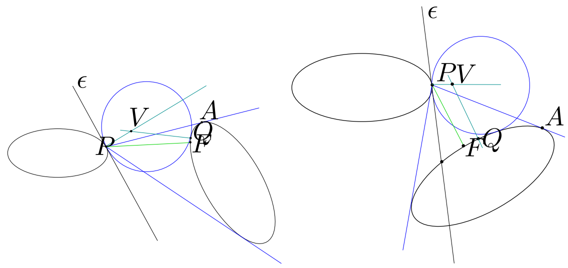


Figure 3: The two cases for defining an Apollonius arc

at r_1 and r_2 are tangent to E_t at points t_1, t_2 respectively. Since t lies between t_1 and t_2, there exists some point r between r_1 and r_2 that corresponds to the Apollonius circle tangent to r and t of E_r and E_t respectively. □

The visible arc may also include some other bitangent circles *internally* tangent to E_t. The subset of the visible arc that contains only the Apollonius circle is called an *Apollonius arc.*

Lemma 5.3 *Given is a point $P = (x(t), y(t))$ on E_t. Consider the line ϵ, tangent at P (cf. fig. 3). If ϵ does not intersect E_r, then the visible arc is an Apollonius arc. Otherwise, the endpoints of the Apollonius arc are: the intersection of ϵ with E_r and the endpoint of the visible arc which lies on the opposite side of E_t with respect to ϵ.*

Proof. If ϵ does not intersect E_r, then it leaves each ellipse in a different hyperplane. In this case, a circle internally tangent to E_t at t, cannot be tangent to E_r as well. Thus, according to the visibility property, the visible arc is an Apollonius arc. If ϵ intersects E_r, then a circle internally tangent to E_t at t can be tangent to E_r at a point that lies in the same hyperplane of ϵ as E_t. Therefore, only the part of the visible arc of E_r that lies in the opposite hyperplane is an Apollonius arc. □ □

We thus computed arc (r_1, r_2) or $(r_1, \infty) \cup (-\infty, r_2)$ on E_r which contains *only* the tangency point of the Apollonius circle, *isolating* it from the tangency points of non-external bitangent circles.

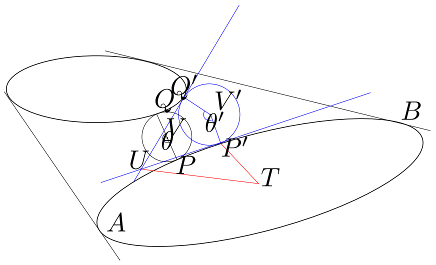


Figure 4: The radius of the Apollonius circle as we move along the boundary

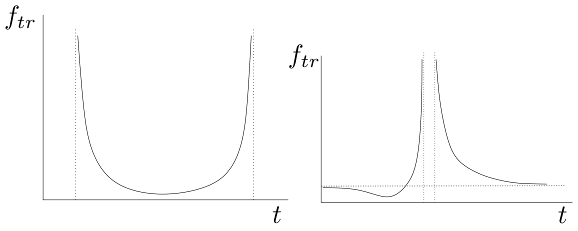


Figure 5: A sample graph of f

Corollary 5.4 *Given a point t_0 on E_t, it is possible to determine the unique root r_i of $B(t_0, r)$, from equation (9), which lies on the Apollonius arc of E_r with respect to t_0.*

Given a point $(x(t), y(t))$ on E_t, the *squared radius* of the Apollonius circle of E_t, E_r tangent to E_t at that point is denoted by $f_{tr}(t)$. From the above, it follows that: $f_{tr}(t) := \big(v_1(t,\hat{r}) - x(t)\big)^2 + \big(v_2(t,\hat{r}) - y(t)\big)^2$, where $\hat{r}$ is the root of (9) that corresponds to the Apollonius circle, when we fix t. Thus,

$$f_{tr}(t) = \frac{1}{4} P_t(t) \left(\frac{A_{tr}(t,\hat{r})}{(1+t^2)(1+\hat{r}^2) D_{tr}(t,\hat{r})} \right)^2 . \quad (10)$$

In the above equation, $P_t(t)$ has no real roots, A_{tr} is a bivariate polynomial of degree 2 in t and 4 in r and $D_{tr} \neq 0$, unless the normals are parallel.

In the sequel, we assume that $f_{tr}(t)$ is defined on a continuous interval (a, b) (left-hand side in fig. 5). If the interval is of the form $(-\infty, a) \cup (b, \infty)$ (right-hand side of fig. 5), then the problem is identical or easier.

Lemma 5.5 *Function $f_{tr}(t)$ consists of two strictly monotone parts, one decreasing and one increasing.*

Proof. Although the proof can be intuitive, we provide a more formal one. There exist two single points P and Q on E_t and E_r respectively whose distance is minimal (fig. 4). As we move from P to P' (in CW orientation), we have: $|\overrightarrow{PV}| + |\overrightarrow{VQ}| < |\overrightarrow{P'V'}| + |\overrightarrow{V'Q'}|$, since $\overrightarrow{P'V'}$ and $\overrightarrow{V'Q'}$ cross the tangent lines at P and

Q respectively and angle $\theta' = \widehat{P'V'Q'}$ is smaller than $\theta = \widehat{PVQ}$. Therefore the radius of the Apollonius circle grows monotonically to infinity. The same arguments can be used to show that the radius also grows when we move in CCW orientation. □

Fig. 5 shows a graph of f. The figure is correct in terms of the function's monotony. We have not proven the function's convexity, though this is suggested by numerical examples. To compute a value of $f_{tr}(t)$ at a given point t we have to determine $\hat{r}$. First, we compute a proper Apollonius arc (r_1, r_2) on the second ellipse. This is an isolating interval of $B(t,r)$ that contains root $\hat{r}$ which corresponds to the Apollonius circle. Now we can compute f_{tr} from equation (10).

5.1 The tritangent circle

In the parametric space, the intersection of two bisectors involves 3 variables, so in order to express the Voronoi circle, we need the intersection of three bisectors. The system $B(t,r) = B(t,s) = B(r,s) = 0$ has a non-optimal mixed volume of 432. We were not able to factor its resultant. We solved this system with the SYNAPS package of multivariate Bernstein subdivision in 3 msec to 1 min, depending on how large the initial domain was.

As an alternative, we consider the system:

$$Q(t,r,s) = B(t,r) = B(t,s) = 0 \qquad (11)$$

Here, Q is the condition that makes the three normals of each ellipse intersect at a single point. Q is a polynomial of total degree 12, 4 in each variable t, r, s. Again, this system has a mixed volume of 432. But now, we construct a resultant matrix whose determinant is factored: We have proven that there are factors which have no real roots, or their roots correspond to the normals lying on the same line. In every example we have tried, if we eliminate all these factors at appropriate powers, we obtain a polynomial of optimal degree (184) that contains all relevant roots. We conjecture that this is the general case. Currently, we have proven that the factors exist, but not with such exponent.

For (11), PHCpack took about a minute to solve with 10-bit coefficients, two times slower than in the implicit approach. We now turn to a subdivision method which is faster. We actually solve only for t, but we show that if we know t with sufficient precision, then we are able to answer κ_3. The idea is that we slide an Apollonius circle tangent to two ellipses, trying to make it tangent to the third one as well. This is implemented by successively smaller intervals on the first ellipse, which define the point of tangency on this ellipse and, moreover, allow us to compute the points of tangency on the other ellipses.

The Voronoi circle is the circle which is externally bitangent to E_t, E_r, E_s at the same time. The tangency point of the Voronoi circle on E_t can be defined by the condition:

$$\mathcal{S}_{trs}(t) = 0, \text{ where } \mathcal{S}_{trs}(t) = f_{tr}(t) - f_{ts}(t).$$

We factor this polynomial as follows:

$$\mathcal{S}_{trs}(t) = \frac{P_t(t)(Q_1 - Q_2)(Q_1 + Q_2)}{4\big[(1+t^2)(1+r^2)(1+s^2)D_{tr}(t,r)D_{ts}(t,s)\big]^2} \qquad (12)$$

where Q_1 and Q_2 are functions of (t,r,s).

We use a customized bisection to find a root of $\mathcal{S}_{trs}(t)$. We only need to solve $Q_1 - Q_2$ and $Q_1 + Q_2$, since the rest of the terms in (12) are always positive.[6] In fact, we only solve $Q_1 - Q_2$, because $Q = (Q_1 - Q_2)/(1 + w_1^2)$. However, we still need $Q_1 + Q_2$ to determine the sign of $\mathcal{S}_{trs}$.

Now we determine starting intervals for the subdivision. Consider the complements ϵ_1, ϵ_2 of the supporting hyperplanes of the two external bitangents of two ellipses E_1, E_2 and a query ellipse E that does not intersect the other two. Let $|\epsilon_i| = 0$ or 1 depending on whether $E \cap \epsilon_i = \emptyset$ or not. Let C be the interior of the convex hull of E_1, E_2. Then, the number of Voronoi circles is $|\epsilon_1| + |\epsilon_2|$, if $E \cap C = \emptyset$, or $2 - |\epsilon_1| - |\epsilon_2|$, otherwise.

So, we can find a starting interval that contains the tangency point of the Voronoi circle. In the case where two Voronoi circles exist, we assume that we know in advance which one we want and therefore we pick a proper sub-interval. We end up with an interval that contains only one Voronoi circle, and hence $\mathcal{S}_{trs}$ has a unique root in the starting interval.

We express the Voronoi circle between E_t, E_r, E_s implicitly by an interval containing t, such that $(x(t), y(t))$ is the tangency point on E_t. Note that this interval might contain tangency points of other non-external tritangent circles. We start by the initial interval $[a,b]$ that contains the tangency point of the Voronoi circle and later, if necessary, we subdivide this interval by bisection. Let $\mathcal{A}_{trs}$ denote an enclosing interval $[a,b]$ where the tangency point on E_t of the Voronoi circle of E_t, E_r and E_s lies. The subdivision operator $(*)$ is defined as follows:

$$\mathcal{A}_{trs} * \mathcal{A}_{trs} := \begin{cases} [\frac{a+b}{2}, \frac{a+b}{2}], & \text{if } \mathcal{S}_{trs}(\frac{a+b}{2}) = 0, \\ [a, \frac{a+b}{2}], & \text{if } \mathcal{S}_{trs}(a)\mathcal{S}_{trs}(\frac{a+b}{2}) < 0, \\ [\frac{a+b}{2}, b], & \text{otherwise.} \end{cases}$$

We denote multiple subdivisions by the power operator. $(\mathcal{A}_{trs})^k$ represents an interval $[a,b]$ that has been subdivided k times and its length is $2^{-k}(b-a)$.

[6] These two polynomials appear because we took the difference of squared functions.

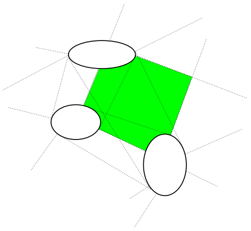


Figure 6: Starting intervals for t, r, s (and region of the Voronoi vertex)

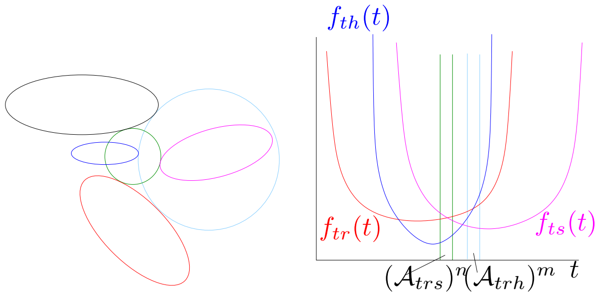


Figure 7: Deciding κ_3

5.2 Deciding κ_3

This subsection shows how the above algorithm decides κ_3 and establishes its exactness.

Given ellipses E_t, E_r, E_s we want to determine the relative position of ellipse E_h with respect to the Voronoi circle of the first 3. The answer of κ_3 is FALSE, TRUE, or 0, depending on whether E_h is outside, intersects the open Voronoi disk, or is externally tangent to the Voronoi circle of E_t, E_r, E_s.

Lemma 5.6 *Let $x \in [a, b]$ be the root of $\mathcal{S}_{trs}(x)$. If $\mathcal{S}_{trh}(x) > 0$, then E_h intersects the Voronoi circle of the other 3 ellipses. If $\mathcal{S}_{trh}(x) < 0$, then E_h lies outside the Voronoi circle. Otherwise, E_h is externally tangent to this circle.*

Proof. If $\mathcal{S}_{trh}(x) > 0$, then there exists a bitangent circle of E_t, E_h tangent at point x of E_t, which lies inside the Voronoi circle of E_t, E_r, E_s. Therefore, E_h intersects the Voronoi circle. If $\mathcal{S}_{trh}(x) < 0$, then the external bitangent circle of E_t on x and of E_h contains the Voronoi circle. Therefore, E_h lies outside the Voronoi circle. If $\mathcal{S}_{trh}(x) = 0$, the two Voronoi circles coincide; this is a degerate configuration. □

Observe that there is a neighborhood $\mathcal{U}$ of x where $\operatorname{sgn}(S_{trh}(u)) = \operatorname{sgn}(S_{trh}(x))$, $\forall u \in \mathcal{U}$. In our implementation, to find $\mathcal{U}$, it suffices that we have separated the roots of $\mathcal{S}_{trs}, \mathcal{S}_{trh}$. Fig. 7 shows an example where the query ellipse intersects the Voronoi circle.

We now establish the *exactness* of our algorithm, by computing the number of bits that suffice in order to certify the predicate. We shall use the system from sec. 4, defined after (8), because it has optimal mixed volume: $\Delta_1(v_1, v_2, q) = \Delta_2(v_1, v_2, q) = \Delta_3(v_1, v_2, q) = q - v_1^2 - v_2^2 + s = 0$. Let us eliminate v_1, v_2, q; the resultant $R(s)$ is of degree 184 in s and has coefficient bit size $3 \cdot 56 \cdot \tau_\Delta = 168\tau_\Delta$ [11]. Here 56 equals the mixed volume of the system $\Delta_i, \Delta_j, q - v_1^2 - v_2^2 + s$, if we consider s as a parameter, and τ_Δ denotes the bit size of the coefficients of Δ_i, where $1 \le i, j \le 3$ and $i \ne j$.

The minimum distance between two roots of a polynomial P (i.e. *separation bound*) of degree d and bit size τ is $sep(P) \ge d^{-(d+2)/2}(d+1)^{(1-d)/2}2^{\tau(1-d)}$ [27], thus the number of bits that we need in order to compute s is no more than $1389 + 30744\,\tau_\Delta$.

In order to compare two radii s_1 and s_2, which are roots of polynomials R_1 and R_2 respectively, we need a bound for $|s_1 - s_2|$. Notice that $|s_1 - s_2| \ge sep(R_1R_2)$, where the polynomial R_1R_2 has degree 368, since we multiply two polynomials of degree 184, and coefficient bit size $8 + 336\tau_\Delta$. The latter follows since we multiply two polynomials of bit size $168\tau_\Delta$, so their product has, in the worst case, a coefficient of magnitude $184 \cdot 2^{2 \cdot 168\tau_\Delta}$, or of bit size $\lceil \lg 184 + 2 \cdot 168\tau_\Delta \rceil$. We conclude [27] that the number of bits sufficient to compare two roots of R_1 and R_2 and thus to compare the two radii s_1 and s_2 is $1508 + 30324\tau_\Delta$, which corresponds to $sep(R_1R_2)$ divided by 2.

This bound is close to tight, since the polynomials R_1 and R_2 are obtained as resultants of systems with optimal mixed volume, thus their degree is 184 and they are irreducible in the general case. Moreover the separation bound is tight (up to some constants), namely the worst case separation bound can be attained by Mignotte's polynomials [27].

If the ellipses are given parametrically, in order to compute the implicit representation (3), the bit size increases by a factor of 6. If the input coefficients have τ bits, then $\tau_\Delta = 6\tau$. If the order of convergence of our method is ϕ, then the number of iterations needed for κ_3 is $\log_\phi(1508 + 181944\tau)$.

6 Conflict region type (κ_4)

This predicate takes as input ellipses E_t, E_r, E_s, E_h and determines the type of conflict of ellipse E_q with the Voronoi edge whose vertices are the centers of the Voronoi circles defined by E_t, E_r, E_s and E_t, E_r, E_h respectively. The conflict region is the set of points V on the Voronoi edge where an Apollonius circle of E_t, E_r centered at V intersects E_q, and it may fall into one of 6 cases [12, 20]: NoConflict, Interior, 123-Vertex, 124-Vertex, TwoVertices, EntireEdge (fig. 8).

As in the previous section we estimate t_3, t_4, such that $t_3 \in \mathcal{A}_{trs}$, $t_4 \in \mathcal{A}_{trh}$ be the tangency points of the two Voronoi circles and assume $t_3 < t_4$. Then, table 1 shows how we can decide κ_4 by applying κ_3. The case $t_3 > t_4$ is treated symmetrically.

The (degenerate) case where $t_3 = t_4$ implies there is a unique Voronoi circle tangent to the 4 given ellipses, which can be detected by κ_3. Then, the Voronoi edge degenerates to a vertex and κ_4 reduces to κ_3 with arguments any 3 of the 4 ellipses and E_q. The possible outcomes are NoConflict, EntireEdge.

case 1,2 E_q is outside both Voronoi circles. It lies in the region between the Voronoi circles and ellipses E_t, E_r, iff the tangency points of the Voronoi circles (there are two of them) of E_t, E_r, E_q lie in $[t_3, t_4]$ (Interior). Otherwise, E_q does not conflict with the bisector (NoConflict). Note that it is not possible to have both κ_3 equal to 0.

case 3,4 The query ellipse conflicts with only one of the Voronoi circles, namely E_t, E_r, E_s (or E_t, E_r, E_h). The center of the (unique) Voronoi circle of E_t, E_r, E_q lies on the Voronoi edge, therefore the predicate answers 123-Vertex (or 124-Vertex), respectively.

case 5,6 E_q conflicts with both Voronoi circles. It has no common points with the region between E_t, E_r, E_s, E_h, iff the tangency points of the Voronoi circles of E_t, E_r, E_q lie in $[t_3, t_4]$. In this case, there is a part of the Voronoi edge that does not conflict with E_q (TwoVertices). Otherwise, E_q conflicts with the entire edge (EntireEdge).

7 Exact implementation

We report on our implementations in C++ and Maple, and illustrate them with a series of experiments. We offer some comparison with existing generic algebraic software for κ_3. All tests ran on a P4 2.6GHz-CPU with 1GB of RAM, using Debian Linux with a 2.6.10 kernel.

We have implemented predicates κ_1, κ_2 in C++ using the specialized algorithms of [13, 14], which are implemented in the SYNAPS library [24].

The real algebraic numbers are in isolating interval representation, that is by a square free integer polynomial and an interval with rationals endpoints. Since predicates κ_1 and κ_2 involve computations with real algebraic number of degree up to 4 we used the implementation based on the algorithms of [13], which avoids subdividisions both for real solving and comparison.

We used extended integer arithmetic from gmp. We also performed tests with the CGAL filtered type Lazy_exact_nt, but the results were not better. The reason is that the size of the various quantities is rather large and the filter almost always failed. This implies that *geometric* filters may be used, cf. sec. 8.

For each test we randomly generated 1000 instances (point and 2 ellipses, or 3 ellipses), with the coefficients uniformly distributed between 1 and 2^B, $B \in \{10, 30, 100, 300\}$. Table 2 summarizes average timings; for κ_1 and κ_2 runtimes grow subquadratically in B. Note that for κ_2 half of the time is spent for the solution of the bivariate system and the other half is spent for the computation of the the relative position of the third ellipse.

B	predicate κ_1[ms]	predicate κ_2[ms]
10	0.45	6.15
30	0.94	16.46
100	3.68	73.21
300	17.3	396.82

Table 2: Timings from our implementation in C++ utilising the implicit approach

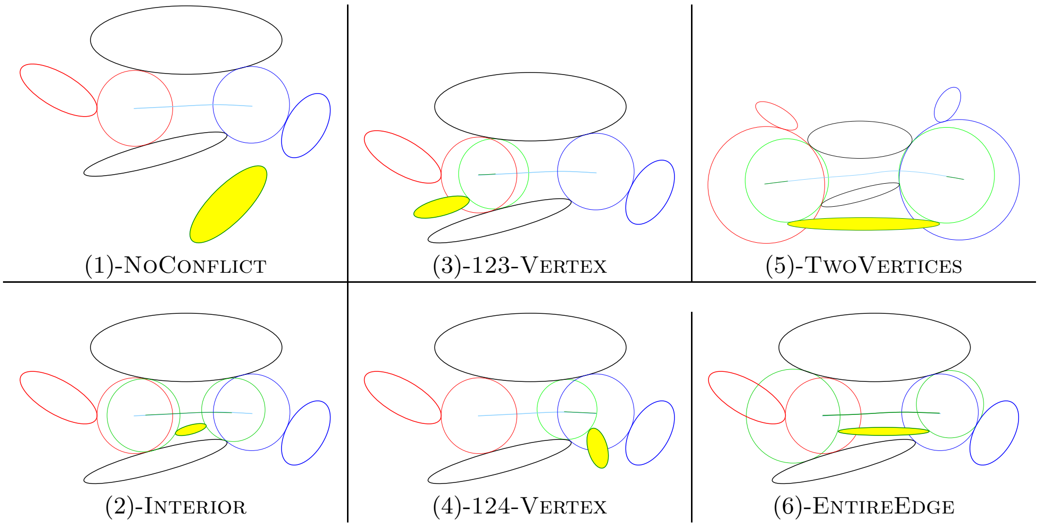
The implementation of κ_3 with parametric ellipses was done in Maple 9. We have implemented a small algebraic number package that performs exact univariate real root isolation, comparison and sign evaluation of univariate (bivariate) expressions over one (two) algebraic number(s), using Sturm sequences and interval arithmetic over $\mathbb{Q}$.

In order to decide the *degenerate cases* we have to go up to the separation bound. This turned out to be quite impractical, because of the number of iterations needed. We suspect a degeneracy if after a certain number of iterations, our algorithm has not yet decided the predicate. Then, given the ellipses E_t, E_r, E_s and the query one E_h, we compute the resultants of system (11) $R_1(t), R_2(t)$, with respect to the two triplets t, r, s and t, r, h. If all four ellipses share a common Voronoi circle, then R_1 and R_2 have a common root. In this case $G(t) = \gcd(R_1, R_2) \neq 1$. Given R_1, R_2 and G, we isolate the roots of the polynomials and run our algorithm with a better separation bound. This turned out to work very well in practice. With 10-bit coefficients, the computation of the resultant of (8) takes about 5 min with interpolation, while of (11) it is computed within 10 sec with two applications of Sylvester resultants. This shows that the parametric system, not only provides a way to answer κ_3 with geometric arguments, but also allows faster computation of the resultant.

We performed several preliminary experiments with different triplets of ellipses and circles. We consider a query ellipse (or circle) with its centre moving along a line and measure the time taken by κ_3 to decide its relative position with respect to the Voronoi

case	$\kappa_3(E_t, E_r, E_s, E_q)$	*condition*	$\kappa_3(E_t, E_r, E_h, E_q)$	κ_4
1	$0 \vee$ False	$(\mathcal{A}_{trq})^k \cap [t_3, t_4] = \emptyset$	$0 \vee$ False	NoConflict
2		$(\mathcal{A}_{trq})^k \subseteq [t_3, t_4]$		Interior
3	True	$(\Rightarrow (\mathcal{A}_{trq})^k \subseteq [t_3, t_4])$	$0 \vee$ False	123-Vertex
4	$0 \vee$ False	$(\Rightarrow (\mathcal{A}_{trq})^k \subseteq [t_3, t_4])$	True	124-Vertex
5	True	$(\mathcal{A}_{trq})^k \subseteq [t_3, t_4]$	True	TwoVertices
6		$(\mathcal{A}_{trq})^k \cap [t_3, t_4] = \emptyset$		EntireEdge

Table 1: Deciding κ_4


(1)-NoConflict

(3)-123-Vertex

(5)-TwoVertices

(2)-Interior

(4)-124-Vertex

(6)-EntireEdge

Figure 8: All cases of κ_4, where the query ellipse is shaded

circle. Among the various configurations, there were both degenerate and non-degenerate cases, although the former are very hard to generate for ellipses.

The ellipses have 10-bit coefficients in their parametric form. Fig. 9 shows the 3 ellipses (or circles) and the query one in its initial, middle and final position. In fig. 10 we present the times for 3 test suites: The first two graphs involve ellipses that do not share a common Voronoi circle with the query one, which center moves along the line $y = -x$. Notice that the time increases as we approach a degenerate configuration. Although the hardest cases took about 5s, in 90% of the cases we can decide in less than 2.5s. The third graph involves circles, but as the query circle moves along $y = 0$, a degenerate configuration is attained. In that case the algorithm has to compute the resultants. This corresponds to the peak of the graph which is 30s after 100 iterations. In all other cases the timings are less than 3s.

Our implementations are exact but can also run with any prescribed precision, e.g. for rendering purposes. In particular, a much faster execution is possible for the above algorithms if we restrict ourselves to machine precision, as in [18]. Fig. 11 presents experiments using machine precision and 32 bit coefficients. With this inexact approach, we can decide the predicate in less than two seconds in all cases.

8 Future work

We are implementing our methods in C++, using Maple as a testbed of ideas. Our final goal is a CGAL implementation. Working in C++ will allow us to use one of the powerful interval arithmetic packages in C++.

We can speed up the subdivision process by noticing that $\mathcal{S}_{trs}$ is strictly monotone in the starting interval $[a, b]$ and that it has a unique simple real root in it. Since we can not prove its convexity we can not guarantee the convergence of a Newton-like subdivision, which has quadratic convergence rate, but we can use Brent's (also known as Van Wijngaarden-Dekker-Brent's) method which is superlinear [9]. Let $[a, b]$ be an interval during the subdivision process and $m = \frac{a+b}{2}$. The new endpoint, i.e. x, is given by the following iterative scheme $x = m + \frac{P}{Q}$, where $R = \frac{\mathcal{S}_{trs}(m)}{\mathcal{S}_{trs}(b)}$, $S = \frac{\mathcal{S}_{trs}(m)}{\mathcal{S}_{trs}(a)}$, $T = \frac{\mathcal{S}_{trs}(a)}{\mathcal{S}_{trs}(b)}$,

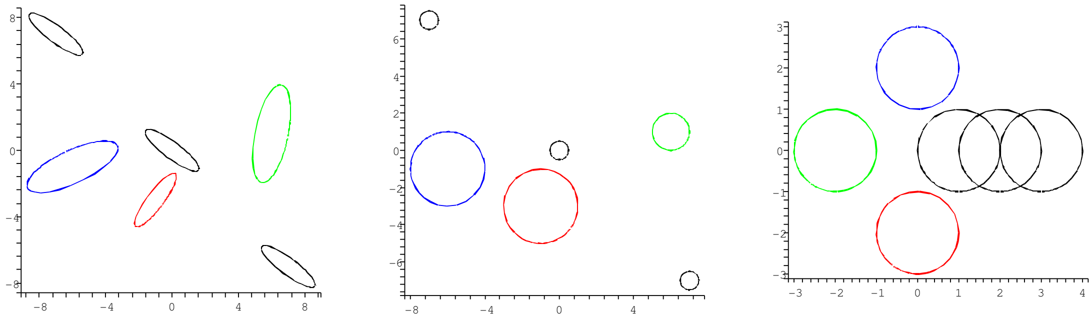


Figure 9: Tested configurations of 3 ellipses (circles) and sample positions of the query object

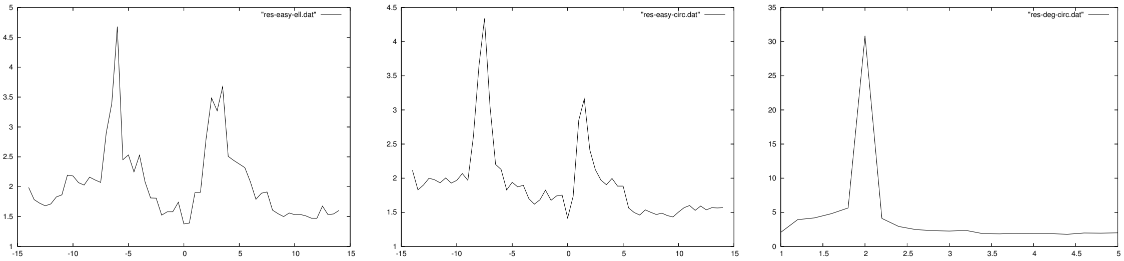


Figure 10: Execution time of κ_3 as function of the position of the query ellipse's (circle's) center

$P = S(T(R-T)(b-m)-(1-R)(m-a))$ and $Q=(T-1)(R-1)(S-1)$. The term $\frac{P}{Q}$ corresponds to a correction factor. If $x \notin [a,b]$ then the new estimation is m. This method is a combination of bisection and inverse quadratic interpolation (similar to the secant method) and ensures that the new estimation will always lie between the interval $[a,b]$. The convergence rate of the method is $\phi = 1.618$ in general. Superlinear convergence is important for obtaining the number of bits required in the degenerate cases.

It is possible to certify our algorithm based on constructive root separation bounds e.g. [10, 26], which should be tighter than the static bounds now used.

We conclude with geometric constraints that permit to derive bounds for the Voronoi vertex, thus speeding up the subdivision.

Theorem 8.1 *Circle $C(V,\sqrt{s})$ is externally tangent to a given ellipse iff the coefficients of $\varphi(\lambda)$ satisfy $\Delta = 0$ and one of the following: Either V lies outside the closed disk of $T(x,y)$, defined in sec. 3, or V lies inside the closed disk $T(x,y)$ and $E(v_1,v_2) > J_1 s$. The latter means, for fixed s, that V is outside the ellipse $E - J_1 s = 0$, which has the same foci as E but different axes.*

Proposition 8.2 *Consider two ellipses E_1, E_2 and point V outside of both. Let $\delta_1 < \delta_2$ be the distances between V and the ellipses. Then the center of an external bitangent circle does not lie in circle $C(V, \frac{\delta_2-\delta_1}{2})$.*

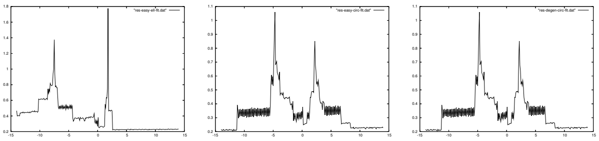


Figure 11: Execution time of κ_3 as function of the position of the query ellipse's (circle's) center, using machine precision

Acknowledgements. The authors would like to acknowledge D. Lazard and S. Petitjean at the 2nd ARCADIA Workshop for their useful comments and suggestions.

References

[1] H. Alt, O. Cheong, and A. Vigneron. The Voronoi Diagram of Curved Objects. *Discrete and Computational Geometry*, 34(3):439–453, Sep 2005.

[2] P. Angelier and M. Pocchiola. A sum of squares theorem for visibility. In *Procs of 17th SoCG*, pages 302–311. ACM Press, 2001.

[3] F. Anton. *Voronoi diagrams of semi-algebraic sets.* PhD thesis, The University of British Columbia, January 2004.

[4] F. Anton, J.-D. Boissonnat, D. Mioc, and M. Yvinec. An exact predicate for the optimal construction of the additively weighted Voronoi diagram. In *Europ. Workshop Comput. Geom.*, 2002.

[5] D. Attali and J.-D. Boissonnat. Complexity of the delaunay triangulation of points on polyhedral surfaces. *Discr. & Comp. Geometry*, 30(3):437–452, 2003.

[6] I. Boada, N. Coll, and J.A. Sellarès. Multiresolution approximations of generalized Voronoi diagrams. In *Proc. Intern. Conf. Comp. Science*, pages 98–106, 2004.

[7] J.-D. Boissonnat and C. Delage. Convex hull and Voronoi diagram of additively weighted points. In *Proc. 13th Annu. European Sympos. Algorithms*, volume 3669 of *Lecture Notes Comput. Sci.*, pages 367–378. Springer-Verlag, 2005.

[8] J.-D. Boissonnat and M. Karavelas. On the combinatorial complexity of Euclidean Voronoi cells and convex hulls of d-dimensional spheres. In *Proc. SODA*, pages 305–312, 2003.

[9] R. Brent. *Algorithms for Minimization without Derivatives.* Prentice-Hall, Englewood Cliffs, N.J., 1973.

[10] C. Burnikel, S. Funke, K. Mehlhorn, S. Schirra, and S. Schmitt. A Separation Bound for Real Algebraic Expressions. In *ESA*, volume 2161 of *LNCS*, pages 254–265. Springer, 2001.

[11] D. Cox, J. Little, and D. O'Shea. *Using Algebraic Geometry.* Number 185 in GTM. Springer, New York, 2nd edition, 2005.

[12] I.Z. Emiris and M.I. Karavelas. The predicates of the Apollonius diagram: algorithmic analysis and implementation. *Comp. Geom.: Theory & Appl., Spec. Issue on Robust Geometric Algorithms and their Implementations*, 33(1-2):18–57, 2006.

[13] I.Z. Emiris and E.P. Tsigaridas. Computing with real algebraic numbers of small degree. In *Proc. ESA*, LNCS, pages 652–663. Springer, 2004.

[14] I.Z. Emiris and E.P. Tsigaridas. Real solving of bivariate polynomial systems. In V.G. Ganzha, E.W. Mayr, and E.V. Vorozhtsov, editors, *Proc. Computer Algebra in Scientific Computing (CASC)*, LNCS, pages 150–161. Springer Verlag, 2005.

[15] I.Z. Emiris and G.M. Tzoumas. Algebraic study of the Apollonius circle of three ellipses. In *Proc. Europ. Works. Comp. Geom.*, pages 147–150, Holland, 2005. Also: Poster session, CASC'05, Greece. To appear in *SIGSAM Bulletin.*

[16] F. Etayo, L. Gonzalez-Vega, and N. del Rio. A new approach to characterizing the relative position of two ellipses depending on one parameter. *Comp.-Aided Geom. Design*, 2005. Submitted.

[17] L. Habert. Computing bitangents for ellipses. In *Proc. 17th Canad. Conf. Comp. Geom.*, pages 294–297, 2005.

[18] I. Hanniel, R. Muthuganapathy, G. Elber, and M.-S. Kim. Precise Voronoi cell extraction of free-form rational planar closed curves. In *Proc. 2005 ACM Symp. Solid and phys. modeling*, pages 51–59, Cambridge, Massachusetts, 2005.

[19] P. Harrington, C.O. Dúnlaing, and C. Yap. Optimal Voronoi diagram construction with n convex sites in three dimensions. Tech. Rep. TCDMATH 04-21, School of Mathematics, Dublin, 2004.

[20] M.I. Karavelas and M. Yvinec. Voronoi diagram of convex objects in the plane. In *Proc. ESA*, pages 337–348, 2003.

[21] D.-S. Kim, D. Kim, and K. Sugihara. Voronoi diagram of a circle set from Voronoi diagram of a point set: II. Geometry. *CAGD*, 18:563–585, 2001.

[22] R. Klein, K. Mehlhorn, and S. Meiser. Randomised incremental construction of abstract Voronoi diagrams. *Comput. Geom. Theory & Appl.*, 3(3):157–184, 1993.

[23] M. McAllister, D. Kirkpatrick, and J. Snoeyink. A compact piecewise-linear Voronoi diagram for convex sites in the plane. *Discrete Comput. Geom.*, 15:73–105, 1996.

[24] B. Mourrain, J. P. Pavone, P. Trébuchet, and E. Tsigaridas. SYNAPS, a library for symbolic-numeric computation. In *8th Int. Symposium on Effective Methods in Algebraic Geometry, MEGA*, Sardinia, Italy, May 2005. to appear.

[25] W. Wang, J. Wang, and M. Kim. An algebraic condition for the separation of two ellipsoids. *Comp. Aided Geom. Design*, 18:531–539, 2001.

[26] C. Yap. On guaranteed accuracy computation. In F. Chen and D. Wang, editors, *Geometric Computation*, volume 11 of *Lect. Notes Series Comp.* World Scientific, 2004.

[27] C.K. Yap. *Fundamental Problems of Algorithmic Algebra.* Oxford University Press, New York, 2000.