International Journal of Computational Geometry & Applications

Predicates for the Exact Voronoi Diagram of Ellipses under the Euclidean Metric

Ioannis Z. Emiris

Department of Informatics and Telecommunications
National Kapodistrian University of Athens
15784 Panepistimiopolis, Athens, HELLAS
emiris@di.uoa.gr

Elias P. Tsigaridas

Department of Informatics and Telecommunications
National Kapodistrian University of Athens
15784 Panepistimiopolis, Athens, HELLAS
et@di.uoa.gr

George M. Tzoumas

Department of Informatics and Telecommunications
National Kapodistrian University of Athens
15784 Panepistimiopolis, Athens, HELLAS
geotz@di.uoa.gr

Received (4 September 2006)
Revised (revised date)
Communicated by Nina Amenta and Otfried Cheong

This article examines the computation of the Voronoi diagram of a set of ellipses in the Euclidean plane. We propose the first complete methods, under the exact computation paradigm, for the predicates of an incremental algorithm: κ_1 decides which one of two 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 two given ellipses; κ_3 decides the position of a query ellipse relative to a Voronoi circle of three given ellipses; κ_4 determines the type of conflict between a Voronoi edge, defined by four given ellipses, and a query ellipse. The article is restricted to non-intersecting ellipses, but the extension to arbitrary ones is possible.

The ellipses are input in parametric representation, i.e. constructively in terms of their axes, center and rotation. For κ_1 and κ_2 we derive optimal algebraic conditions and provide efficient implementations in `C++`. For κ_3 we compute a tight bound on the number of complex tritangent circles and design an exact symbolic-numeric algorithm, which is implemented in MAPLE. This essentially answers κ_4 as well. We conclude with current work on optimizing κ_3 and on its implementation in `C++`.

Keywords: Voronoi diagram; ellipse; predicate; Euclidean distance; bisector; parametric representation; exact computation

1. Introduction

The Voronoi diagram is one of the essential tools in robot motion planning amidst obstacles. In this problem the computation of a path of the moving object can be transformed to a computation of a path in a Voronoi diagram. Applications, such as navigation among objects, benefit the most from the exact Voronoi diagram of ellipses, since ellipses model different kind of obstacles.

Another motivation comes from visibility problems among ellipses[18] or pairwise disjoint bounded convex sets of constant complexity[2]. In these problems is needed the computation and characterization (as external or internal) of all bitangents of two ellipses; a problem of independent interest. The predicate κ_2 involved in the computation of the Voronoi diagram of ellipses can answer such questions.

In this article we study the Voronoi diagram of ellipses (see fig. 1) under 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. As is the case for almost all problems in computational geometry for curved objects, the algorithm relies heavily on predicates implemented by algebraic operations.

The Voronoi diagram of ellipses can be considered as a generalization of the Apollonius diagram of hyperspheres (sec. 2.4.3 of the ECG book[9]). Another diagram similar to the Voronoi diagram is the Power diagram (sec. 2.3.3 of the ECG book[9]), which simplifies the predicates, but changes the distance function. An extension of this diagram to ellipses should lead to a simpler algorithm. Nonetheless, we use the Euclidean distance as a more natural metric.

We design and implement exact and complete algorithms for the predicates needed in the framework of *abstract* Voronoi diagrams[24] and, more particularly, for the incremental algorithm of Karavelas and Yvinec[22]. 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 provide sufficient information.

Our final goal is a CGAL [a] software package for constructing the Voronoi diagram of ellipses, based on the CGAL implementation for circles[14], 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. McAllister et al[26] compute the diagram of convex polygons, with an approach similar to our subdivision technique, since their algorithm "moves" on the objects' boundary using pruning. Recent efforts have extended Voronoi diagrams to the case where the sites are curves[1,3] or have non-empty interior[8]. In particular, the diagram of circles has been implemented in CGAL [14]; see also the work of Anton et al[4] and Kim et al[23]. Anton[3] examines

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

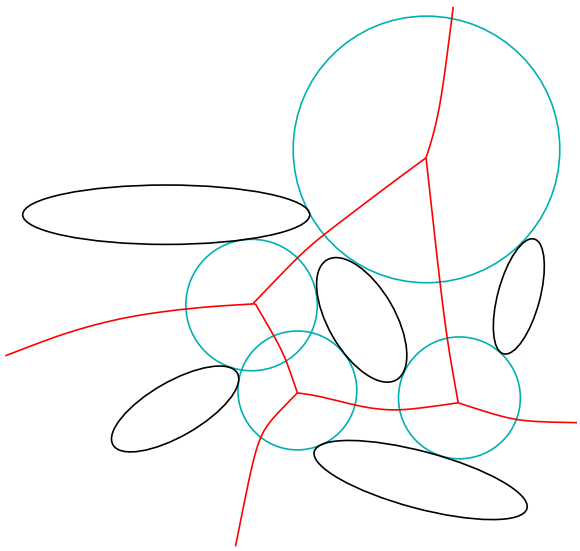

Fig. 1. Voronoi diagram of five ellipses

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.

Harrington et al[20] derive an optimal combinatorial algorithm for constructing Voronoi diagrams of strictly convex rounded sites in $\mathbb{R}^3$, but the predicates are not considered. Boissonnat and Delage[7] describe a dynamic algorithm for constructing the Voronoi diagram of additively weighted points in $\mathbb{R}^d$. This specializes to the Voronoi 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[5]. Boada et al[6] compute a polygonal approximation of a Voronoi diagram at different levels of detail.

Perhaps the work coming closest to ours is the approach of Hanniel et al[19]. 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 follow the exact computation paradigm, but can also run with any prescribed precision.

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 we expect to generalize to arbitrary ellipses and even pseudo-circles[22]. We assume that the input ellipses are given *constructively* in terms of their axes, center and

rotation, all being rational, or if they are given implicitly that we are able to switch representation using only rational arithmetic (see next section).

The four predicates of the incremental algorithm[22] are:

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

Our first contribution are algorithms for κ_1 and κ_2 that are optimal in terms of the degree of the algebraic numbers involved. In fact, for κ_2, we compute and characterize all bitangents of two ellipses using our own tools for dealing with algebraic numbers of degree four. Both algorithms satisfy the requirements of exact computation and are implemented in `C++`.

Using the implicit representation, we obtain the first tight bound on the number of *complex* tritangent circles to three ellipses, namely 184. The number of real tritangent circles remains open. This approach does not lead to an efficient algorithm by itself, however, it provides a nearly optimal theoretical bound on the bit complexity of the problem.

By exploiting the 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 three 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. However, instead of running the subdivision algorithm until the (theoretical) separation bound is reached, our method switches to an exact algebraic solver, after a predescribed number of iterations.

We present and implement in MAPLE a symbolic-numeric algorithm for κ_3. The first step of the algorithm is a customized subdivision-based scheme which "moves" on the border of parametrically defined ellipses, using the parametric representation of the ellipses. Almost all the easy cases are answered during this step. If we are in a degenerate, or near a degenerate configuration, then, after some subdivision steps, we switch to an exact algebraic method in the parametric space which corresponds to real solving a univariate polynomial and comparing real algebraic numbers. Our approach 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.

The algorithm for κ_3 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.

A preliminary version of most of our work appeared in [13].

The article is organized as follows. The next section discusses representation issues. In sections 3 and 4 we give algorithms to decide predicates κ_1 and κ_2. Sec. 5 studies the Voronoi circle from the implicit representation viewpoint. The parametric representation is considered in sec. 6, where we combine the algebraic tools with a subdivision method to yield a symbolic-numeric algorithm for κ_3. Predicate κ_4 is settled in sec. 7. Sec. 8 illustrates our implementations with various tests and finally, the last section concludes with future work.

2. Representation

We assume that an ellipse is given in rational parametric form, i.e. constructively in terms of its rational axes, center and rotation. It is more intuitive for a GUI to offer constructive input rather than require the coefficients of the implicit form. The parametric approach is suitable for tracing the boundary of an ellipse, which is crucial for the subdivision step of the symbolic-numeric algorithm for κ_3. Moreover, given the parametric form it is always possible to derive the implicit one using only rational arithmetic. If the ellipses were given implicitly it is not always possible, using only rational arithmetic to derive the parametric approach.

The parametric representation is

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

where $2\alpha, 2\beta$ are the lengths of the major and minor axes, respectively, $t = \tan(\theta/2) \in (-\infty, \infty)$, θ is the angle that traces the ellipse, $w = \tan(\omega/2)$, ω is the rotation angle between the major and horizontal axes and (x_c, y_c) is its rational center. This representation leaves out of the boundary a single point, called the i-point.

The symmetric ellipse (with respect to its center) is

$$\bar{x}(t) = -x(-t) + 2x_c \text{ and } \bar{y}(t) = -y(t) + 2y_c.$$

We call it the *twin* ellipse. Every point of an ellipse is different from its twin point, including the i-point. We denote by E_t an ellipse parametrized by t and by $\bar{E}_t$ its twin ellipse.

An ellipse has the following implicit equation:

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

The coefficients of (2) are polynomials in the coefficients of (1):

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

Note that χ and ψ express the equations of the major and minor axes, respectively, evaluated at (x_c, y_c). The following quantities are *invariant* under rotation and translation:

$$J_1 = a + c = \alpha^2 + \beta^2, \quad J_2 = ac - b^2 = \alpha^2\beta^2, \tag{4}$$

while $J_4 = J_2(x_c^2 + y_c^2 - J_1)$ is invariant under rotation. Now, the coordinates of the center of the ellipse are

$$(x_c, y_c) = \left(\frac{be - cd}{J_2}, \frac{bd - ae}{J_2}\right). \tag{5}$$

When an ellipse is given in parametric form constructively (rational axes, center and w), the equations in (3) allow us to transform it to its implicit form.

3. Predicate κ_1

For κ_1, we are given two ellipses and a point outside of both, and we wish to find the one closest to the point, under the Euclidean metric. We define the distance from a point to an ellipse as the length of the segment from the point to the ellipse, which is perpendicular to the ellipse.

We start by obtaining a lower bound on the inherent complexity. Take a point V outside an ellipse; it may have up to four normals to the ellipse (fig. 2 left) depending on its position relative to the *evolute* curve, which is a stretched astroid (fig. 2 right). There are four normals if V lies inside the evolute, three normals if V is on the evolute but not at a cusp and two normals if V is on a cusp or outside the evolute[21].

3.1. *The implicit approach*

Consider an ellipse E in the implicit representation as in (2) 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.

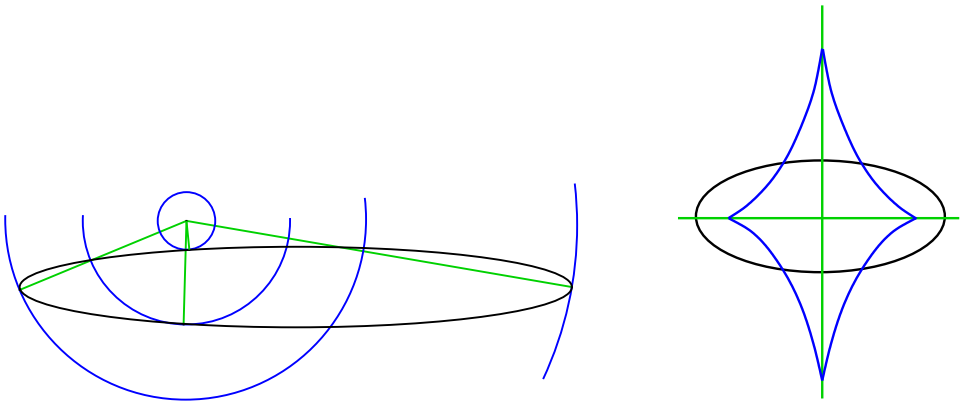

Fig. 2. Left: an example of a point with four normals to a given ellipse. Right: The evolute of an ellipse.

Let us express a conic as $[x, y, 1]M[x, y, 1]^T$, where M is an appropriate 3×3 real symmetric matrix. Then the matrices for E and C are

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

Their *pencil* is $\lambda\,\mathbf{E} + \mathbf{C}$ and their characteristic polynomial is

$$\varphi(\lambda) = \det(\lambda\,\mathbf{E} + \mathbf{C}(s)) = J_2^2\lambda^3 + c_2(s)\lambda^2 + c_1(s)\lambda + s, \tag{6}$$

where J_1, J_2 are defined in (4) and

$$\begin{aligned} c_1(s) &= J_1 s - E(v_1, v_2), \\ c_2(s) &= J_2 s - T(v_1, v_2), \\ T(v_1, v_2) &= J_2[(v_1 - x_c)^2 + (v_2 - y_c)^2 - J_1]. \end{aligned}$$

Notice that $\phi(\lambda)$ is cubic polynomial in λ; its discriminant is:

$$\begin{aligned} \Delta(s) = {} & J_2^2(J_1^2 - 4J_2)\,s^4 + \\ & 2J_2(9J_1J_2^2 - J_1^2T + 6J_2T - 2J_1^3J_2 - J_1J_2E)\,s^3 + \\ & (-18J_2^3E + 4J_1J_2ET - 27J_2^4 + J_1^2T^2 - 18J_1J_2^2T \\ & + J_2^2E^2 + 12J_1^2J_2^2E - 12J_2T^2)\,s^2 + \\ & 2(2T^3 - J_1ET^2 - 6J_1J_2^2E^2 + 9J_2^2ET - J_2E^2T)\,s + \\ & E^2(T^2 + 4J_2^2E), \end{aligned} \tag{7}$$

where $E = E(v_1, v_2)$ and $T = T(v_1, v_2)$. It is interesting that T stands for a circle centered at the center of E with squared radius equal to J_1 (see fig. 3). A circle is externally tangent to an ellipse iff $\varphi(\lambda)$ has a positive double root (and one negative root)[30,17]. Since we want $\varphi(\lambda)$ to have a multiple root its discriminant Δ

must vanish. Notice that from (7), $\Delta(s)$ is a univariate polynomial in s of degree four.

Using this notation, $\delta(V, E)$ is the square-root of the smallest positive zero of $\Delta(s)$. The (algebraic) degree of the coefficients of $\Delta(s)$, in v_1, v_2 and the parameters of E, is six, eight, ten, twelve, and fourteen, in order of decreasing power in s.

If we consider $\Delta(v_1, v_2, s)$ as a trivariate polynomial, then the algebraic degree of its coefficients, which are polynomials in the parameters of E, is six.

Proposition 1. *Given ellipses E_1, E_2 and a point V outside both of them, we can decide which ellipse is closest to V by comparing two algebraic numbers of degree at most four.*

Proof. We consider the polynomials $\Delta_1(s)$ and $\Delta_2(s)$, that are quartic polynomials in s, the smallest positive roots of which correspond to $\delta(V, E_1)$ and $\delta(V, E_2)$, respectively. Thus the distances are algebraic numbers of degree ≤ 4 and the proposition holds. □

In the previous proof it is stated that the distances are real algebraic numbers of degree ≤ 4 and not exactly 4. This is so because, in the presence of multiple roots, we can compute the real roots, almost always, as rational polynomial functions in the coefficients of the polynomial[15].

The degree is optimal with respect to the *degree of the algebraic numbers* involved, when the distances are computed. This follows from the fact that in the worst case a point outside an ellipse has up to four normals and thus in order to compute its distance from the ellipse we must solve a quartic polynomial. Moreover, the arithmetic complexity of real root isolation of a quartic polynomial and comparison of real algebraic numbers of degree ≤ 4 is $\mathcal{O}(1)$[15] thus κ_1 can be answered in constant time. If we are interested in the bit complexity then, if τ is a bound on the bit size of the coefficients, the bit complexity is $\mathcal{O}_B(\tau \lg \tau \lg\lg \tau)$.

Before ending this section, we would like to present some semi-algebraic conditions that allow us to determine whether a circle is externally tangent to an ellipse. These conditions can be derived from the following corollary of Descartes' rule:

Corollary 1. *For a polynomial of degree d, the number of sign variations in its coefficient sequence gives precisely the number of positive roots, assuming no root equals zero and there are d real roots.*

Now, apply this corollary to the characteristic polynomial $\phi(\lambda)$. Its sign sequence is $(+, \text{sign}(c_2), \text{sign}(c_1), +)$, and the constant coefficient is $s > 0$. The sequence of $\phi(-\lambda)$ is $(-, \text{sign}(c_2), -\text{sign}(c_1), +)$. Recall that for the (parametric) circle to be external to the ellipse, $\phi(\lambda)$ must have a double positive root and one negative root.

Lemma 1. *The (parametric) circle is externally tangent to the ellipse, iff the coefficients of $\phi(\lambda)$ satisfy $\Delta = 0$ and one of the following: either $c_2 \geq 0$ and $c_1 < 0$, or $c_2 < 0$ (fig. 3).*

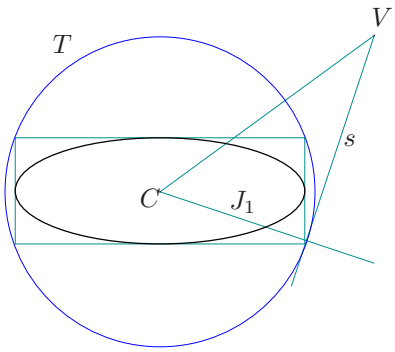

Fig. 3. The geometric meaning of $c_2 < 0$

Proof. [⇐] $\Delta = 0$ ensures a multiple root, and 2 sign variations ensure a double root.
[⇒] External tangency enforces existence of one negative root and one positive double root. Now, $\Delta = 0$ since there is at least one multiple root, hence all 3 roots of ϕ are real; thus, cor. 1 applies. The case we wish to describe corresponds to 2 sign changes for $\phi(\lambda)$ and one for $\phi(-\lambda)$. There are two cases for c_2. If $c_2 \geq 0$, then the sign sequences of $\phi(\lambda), \phi(-\lambda)$ become

$$(+, \geq 0, \text{sign}(c_1), +) \text{ and } (-, \geq 0, -\text{sign}(c_1), +),$$

so we must have $c_1 < 0$. If $\text{sign}(c_2) < 0$, then the sequences are

$$(+, -, \text{sign}(c_1), +) \text{ and } (-, -, -\text{sign}(c_1), +),$$

so anything goes for $\text{sign}(c_1)$. □

The above lead to the following theorem:

Theorem 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.*

3.2. *The parametric approach*

Consider two ellipses E_t, E_r and a point $V(v_1, v_2)$. We may consider V as the intersection of the two normal lines at points t and r on each ellipse, respectively, which are

$$c_x(t)v_1 + c_y(t)v_2 = c,$$
$$d_x(r)v_1 + d_y(r)v_2 = d.$$

Since we have a linear system, the coordinates of V are rational functions in t and r. Let us now compare the distances:

$$[v_1(t,r) - x(t)]^2 + [v_2(t,r) - y(t)]^2 - [v_1(t,r) - x(r)]^2 - [v_2(t,r) - y(r)]^2.$$

Now, in order to decide the predicate we have to perform computations with two algebraic numbers of degree eight. This is so, because the values of the parameters of the two ellipses, for which the tangency points occur are of degree four and thus the distance of a point to an ellipse is expressed by an algebraic number of degree eight. However, we can use (3) to transform the ellipses to the implicit form and decide the predicate as before, using algebraic numbers of degree four and a smaller amount of operations.

In sec. 8 we report on our exact implementation in `C++`.

4. Predicate κ_2

Predicate κ_2 decides the position of a query ellipse relative to one (externally) bitangent line to two given ellipses. The line partitions the plane to two half-planes. We want to know if the query ellipse lies entirely in one of the two half-planes (and which), if it intersects the line or if it is tangent to it. In fact, our algorithm provides additional information, i.e. characterizes all bitangent lines of two ellipses. This may be used to speed up the overall algorithm.

Habert[18] proposes a more complicated approach for determining the type of a single bitangent line without computing all bitangents. He also has to deal with algebraic numbers of degree four but uses a smaller amount of operations. No additional information is computed in this case and although for his problems this is sufficient, it is not clear that in our case we would obtain greater speedup.

For this predicate, the implicit and the parametric approach are of equal complexity, since both require computations with real algebraic numbers of degree up to four. However, in the parametric case we can decide the characterization of all the bitangents after characterizing two of them, while in the implicit case we have to characterize three.

4.1. *The implicit approach*

Consider a (non-vertical) line $L : y - u\,x - v$ and an ellipse E in implicit form, as in (2). The common points of L and E correspond to the solutions of the system $L(x,y) = E(x,y) = 0$. Since $L(x,y)$ is linear, we can solve it for y and substitute its value in $E(x,y)$, obtaining the polynomial

$$R := \left(2\,bu + a + cu^2\right) x^2 + (2\,cvu + 2\,d + 2\,eu + 2\,bv)\,x + f + cv^2 + 2\,ev$$

which is univariate in x. Notice that R is the resultant of the system $L(x,y) = E(x,y) = 0$ with respect to y, that is it expresses the projection of the roots on the x axis. The solutions of $R(x) = 0$ are the abscissas of the intersection points of L and E. Since L and E may have up to two common points, in order L to

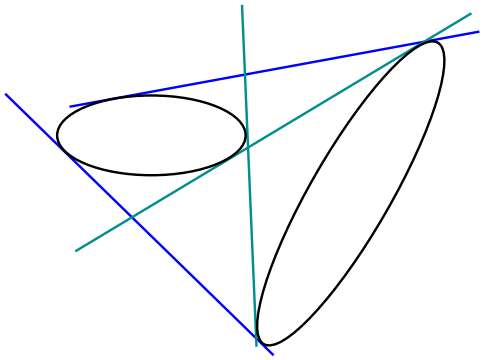

Fig. 4. Four bitangent lines of two ellipses, two external and two internal.

be tangent to E, the common point must be of multiplicity two, as a root of the system. Moreover, $R(x)$ must have one root and since it is of degree two, this root is a double root and so the discriminant of R:

$$\Lambda(u,v) = (e^2-cf)u^2+2(cd-be)uv+(b^2-ac)v^2+2(de-bf)u+2(bd-ae)v-af+d^2$$

must vanish.

Now consider two ellipses E_1 and E_2. We can compute the parameters of the line(s) L tangent to both of them as the solutions of the system

$$\Lambda_1(u,v) = \Lambda_2(u,v) = 0, \tag{8}$$

where the $\Lambda_1 = 0$ (respectively $\Lambda_2 = 0$), expresses the fact that L is tangent to E_1 (respectively E_2). The system is of total degree two, can be solved in $\mathcal{O}(1)^{15}$ and the real solutions are pairs of real algebraic numbers of degree ≤ 4. There are at most four solutions which correspond to four bitangent lines.

It remains to characterize the bitangent lines as internal or external ones. A bitangent line is external if both ellipses are on the same half-plane of the two ones that it defines; otherwise it is internal (fig. 4). Thus, a bitangent line is external iff its equation yields the same sign when evaluated at the center of each ellipse.

Let $\bar{L} : y - \bar{u}\,x - \bar{v}$, be one of (the four possible) bitangent lines, where $(\bar{u}, \bar{v})$ is a solution of system (8) and (x_{c_1}, y_{c_1}), respectively (x_{c_2}, y_{c_2}), is the (rational) center of E_1, respectively E_2, see (5). $\bar{L}$ is an external bitangent if $\text{sign}(\bar{L}(x_{c_1}, y_{c_1}))\cdot \text{sign}(\bar{L}(x_{c_1}, y_{c_1})) > 0$. The signs can be computed directly or we can reduce the computation to comparison of two real algebraic numbers of degree ≤ 4. In either case the complexity is $\mathcal{O}(1)^{15}$. The combination of signs, over all the solutions of the systems and the two ellipses, yields the characterization of the bitangent lines as externals or internals. A vertical bitangent line yields a simpler (univariate) system (8) and can be 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, i.e. $\Lambda_3(\bar{u}, \bar{v})$. Λ_3 is negative, zero, or positive iff E_3 has zero, one or two common points with $\bar{L}$ respectively. In the first two 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.

Notice that all the computations involve real algebraic numbers of degree ≤ 4. This degree is optimal with respect to the degree of the real algebraic numbers involved, when the bitangent lines are computed, since there are up to four bitangents.

4.2. *The parametric approach*

Let us suppose that the ellipses are represented parametrically as in (1). The two ellipses are $E_t(\alpha_t, \beta_t, w_t, x_{c_t}, y_{c_t})$ or E_t, $E_r(\alpha_r, \beta_r, w_r, x_{c_r}, y_{c_r})$ or E_r and the query ellipse is $E_s(\alpha_s, \beta_s, w_s, x_{c_s}, y_{c_s})$ or E_s.

We consider the tangent at point $(x(t), y(t))$ of the 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 (1) we obtain a polynomial of degree two with respect to t. Now, 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. The algebraic degree of the coefficients of $\Lambda_{tr}(t)$ is 12, in the parameters of the two ellipses. The discriminant is of degree four with roots that correspond to the tangency points on E_t of the four bitangent lines of E_t, E_r. We ignore the denominator which is $(1 + w_2^2)(1 + r^2)$.

A bitangent line is external to both ellipses iff its equation yields the same sign when evaluated at an interior point (i.e. center) 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 four.

Lemma 2. *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 $(\mu\epsilon\epsilon\mu)$.*

Proof. Consider a supporting line of the two ellipses, tangent to one of them. As we roll this line around the first ellipse, it will hit all four bitangent lines in $\epsilon, \mu, \mu, \epsilon$ order. Depending on the location of the i-point of the first ellipse with respect to the tangency point of the first external bitangent, we get a different cyclic permutation. □

Corollary 2. *Given two ellipses parametrically, in order to determine the permutation of their bitangents, it suffices to determine the type of exactly two bitangents.*

We now arrive at the following:

Theorem 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 four, over $\hat{t}$, which is a root of $\Lambda_{tr}(t)$ (also of degree four). Now* $\operatorname{sign}(\Lambda_{ts}(\hat{t})) = -1, 0,$ *or* 1 *iff E_s does not intersect, is tangent to, or intersects the bitangent, respectively.*

With the same reasoning as in the implicit case the degree of the algebraic numbers employed is optimal. The same holds not only for the predicate but also for the computation and the characterization of the bitangents of two ellipses.

The predicate is implemented in `C++`, see timings in sec. 8.

5. Implicit approach to κ_3

Given three ellipses E_1, E_2 and E_3, 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, E_0, relative to this circle. This section considers all ellipses in implicit form and applies certain techniques from algebraic geometry for studying the problem. For an introduction to the algebraic notions, the reader may refer to Cox et al[12].

Let $\sqrt{s}$ be the radius of the tritangent circle and (v_1, v_2) its center. For each of the three ellipses and the circle, we consider the discriminant, as in (7), and we get the system

$$\Delta_1(v_1, v_2, s) = \Delta_2(v_1, v_2, s) = \Delta_3(v_1, v_2, s) = 0. \tag{9}$$

The following lemma is quite straightforward.

Lemma 3. *A solution $(\bar{v}_1, \bar{v}_2, \bar{s})$ of system (9) 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 $\bar{s} = s_0^+$.*

Notice that among the solutions of system (9), the external tritangent circle of interest may or may *not* have the smallest radius (fig. 5).

The previous lemma provides a straightforward solution to κ_3, but requires algebraic tools of which there is no efficient and exact implementation as of today. This will become clearer in the sequel, where we study the complexity of the system.

The mixed volume of a polynomial system bounds the number of complex solutions by exploiting the structure (non zero terms) of the equations[12]. The mixed volume of system (9) is 256 and in order to reduce it, we remove solutions at infinity by introducing a new variable q.

$$q = v_1^2 + v_2^2 - s. \tag{10}$$

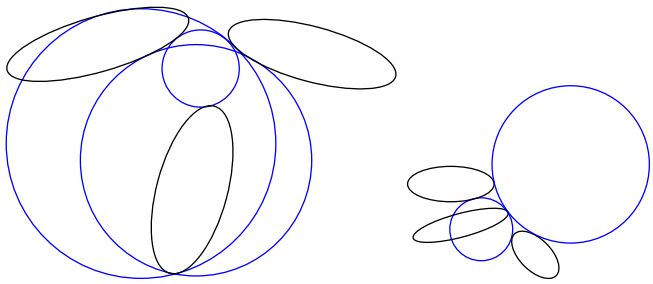

Fig. 5. Tritangent circles to three ellipses; only one is externally tangent

Replacing s by q in system (9) and adding (10) yields the system

$$\Delta_1(v_1, v_2, q) = \Delta_2(v_1, v_2, q) = \Delta_3(v_1, v_2, q) = q - v_1^2 - v_2^2 + s = 0 \qquad (11)$$

with unknowns v_1, v_2, q and s, that has mixed volume 184. If we ignore the fourth polynomial and variable s, the mixed volume is again 184.

The resultant is an irreducible polynomial in the coefficients of an overconstrained system of $n+1$ polynomials in n variables, the vanishing of which is the minimum condition of solvability of the system. By considering only v_1, v_2 and s as unknowns in (11) we have computed the resultant as a polynomial of degree 184 in q.

Each Δ_i is the discriminant of φ_i which is the characteristic polynomial of ellipse i and the Voronoi circle, see (6). The vanishing of Δ_i means that φ_i has a multiple root or equivalently that φ_i and $\frac{\partial}{\partial \lambda_i}\varphi_i$ have a common root, for $i = 1, 2, 3$. Thus an equivalent system to (11), with the same mixed volume, is

$$\varphi_i(v_1, v_2, q, \lambda_i) = \frac{\partial}{\partial \lambda_i}\varphi_i(v_1, v_2, q, \lambda_i) = 0, \text{ for } i = 1, 2, 3. \qquad (12)$$

The bit size of the coefficients of (12) is substantially smaller than that in (11).

Theorem 3. *Three ellipses admit at most 184 complex tritangent circles. This is tight since there are triplets attaining this number.*

The mixed volume provides an upper bound while the degree of the resultant (of our example) gives a lower bound. The theorem generalizes to all types of conics.

Recall[14] that in the case of three circles, the number of tritangent circles is eight and the corresponding predicate is of algebraic degree two. The interesting open question is how many of these circles can be real. Solving system (12) numerically with PHCpack we found up to 22 real solutions (we checked the real roots by hand). The total number of complex solutions was 184. There is a construction[29] 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.

Even though systems (11) and (12) have optimal mixed volume, solving them is a very difficult computational task, even with numerical solvers, let alone exact

ones. To illustrate this we solved system (12) using PHCpack[b] which implements homotopy continuation and is the state-of-the-art in numerical polynomial system solving. For ellipses with coefficients' bit size 30 and 100 the timings of PHCpack were 23.85 and 38.36 seconds, respectively. Anton[3] reports that the time to solve the system describing the Voronoi circle is several minutes, using sparse resultants. We also tried the iCOs interval-arithmetic solver[c] on the system of Δ_i's with coefficients' bit size 60. It detects a degeneracy for κ_3 (three ellipses and a query one, all externally tangent to the same circle) in about 213 seconds on a 1GHz P3. Recently, with some preliminary experiments[25] exploiting the Gröbner base computation, we were able to isolate all real roots of the system in about two minutes with GB-Rs[d].

Notice that solving system (11), or (12), does not answer the predicate since then we must apply Lem. 3. However, since the mixed volume of system (11) is optimal we will use it in sec. 6 in order establish a bit complexity bound for the predicate.

6. Parametric approach to κ_3

We use the parametric representation of ellipses to study the external bitangent circles. In subsec. 6.1 we shall apply this discussion to external tritangent circles.

We express the Voronoi vertex by the intersection of two 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

$$B(t,r) = \big(v_1(t,r) - x(t)\big)^2 + \big(v_2(t,r) - y(t)\big)^2 - \big(v_1(t,r) - x(r)\big)^2 - \big(v_2(t,r) - y(r)\big)^2, \tag{13}$$

which is rational in t, r with denominator D_{tr}. The numerator is a bivariate polynomial of degree six in t and six 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.

[b] http://www.math.uic.edu/∼jan/PHCpack/
[c] http://www-sop.inria.fr/coprin/ylebbah/icos/
[d] http://fgbrs.lip6.fr/salsa/Software/index.php

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

Proof. If we fix t, equation (13) has six complex solutions with respect to r. Therefore, six is an upper bound for the number of possible real bitangent circles. Moreover, a configuration of two ellipses that have six real bitangent circles can be attained, see fig. 6. □

Note that only one such circle is *external* to both ellipses. We call this unique external bitangent circle the *Apollonius* circle of the two ellipses, e.g. the third circle from the right in fig. 6. The Voronoi circle of three ellipses is where three Apollonius circles coincide.

Given ellipses E_t, E_r as in fig. 8, the tangency points of any Apollonius circle lie inside their Convex Hull (CH). Thus, for the parameterization (1), 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*.

Lemma 4. *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. 7), 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 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 one tangency point, namely the one corresponding to the Apollonius circle is called an *Apollonius arc*.

Lemma 5. *Given is a point $P = (x(t), y(t))$ on E_t. Consider the line ϵ, tangent at P (see fig. 7). 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.

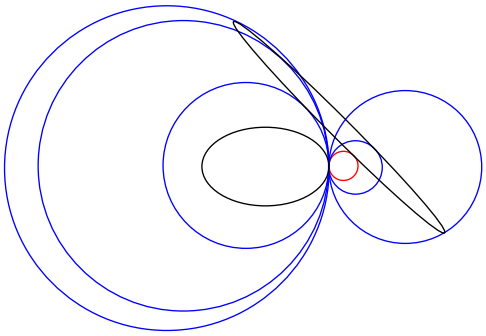

Fig. 6. The six bitangent circles: The Apollonius circle is the 4^{th} from the left

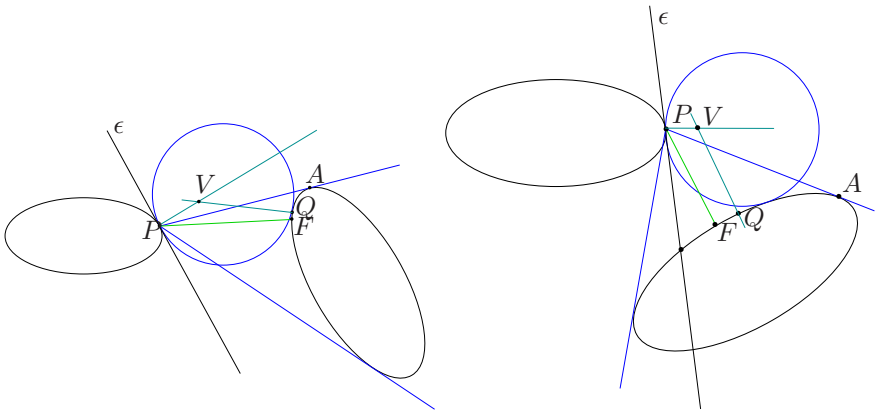

Fig. 7. The two cases for defining an Apollonius arc

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 obtain 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.

Corollary 3. *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 (13), 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

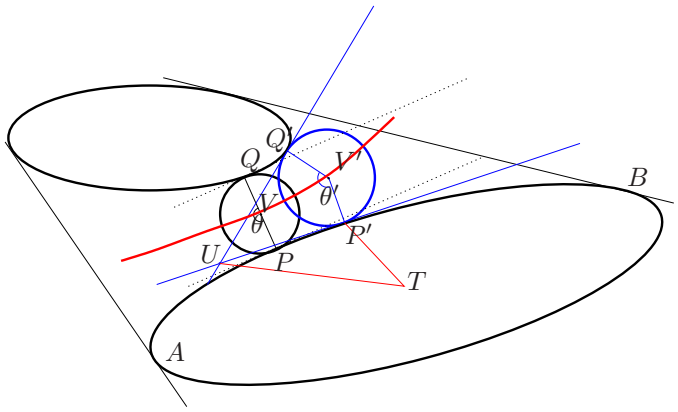

Fig. 8. The radius of the Apollonius circle as we move along the boundary

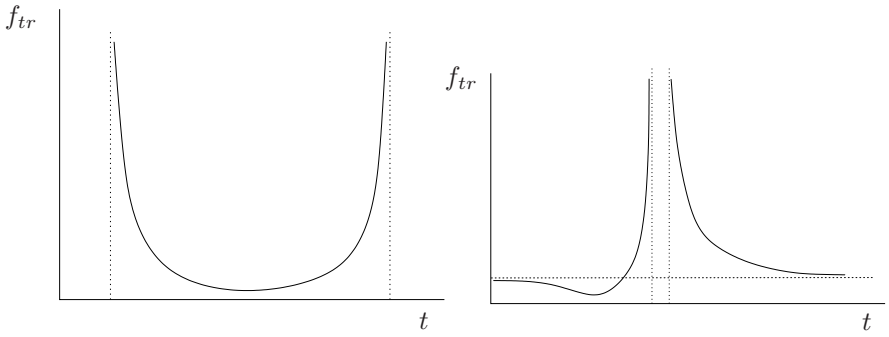

Fig. 9. A sample graph of f

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 (13) that corresponds to the Apollonius circle, when we fix t. Thus,

$$f_{tr}(t) = \frac{1}{4} P_t(t) \left(\frac{A_{tr}(t,\hat{r})}{(1+t^2)(1+\hat{r}^2) D_{tr}(t,\hat{r})} \right)^2 . \tag{14}$$

In the above equation, $P_t(t)$ has no real roots, A_{tr} is a bivariate polynomial of degree two in t and four 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. 9). If the interval is of the form $(-\infty, a) \cup (b, \infty)$ (right-hand side of fig. 9), then the problem is identical or easier.

Lemma 6. *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. 8). 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. 9 (left hand side) 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 $f_{tr}(t)$ from equation (14).

6.1. *The tritangent circle*

In the parametric space, the intersection of two bisectors involves three variables, so in order to express the Voronoi circle, we need the intersection of three bisectors, namely the system $B(t, r) = B(t, s) = B(r, s) = 0$. As a more efficient alternative, we consider the system:

$$Q(t, r, s) = B(t, r) = B(t, s) = 0. \tag{15}$$

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, four in each variable t, r, s. This system has a mixed volume of 432. But, we construct a resultant matrix whose determinant is factored. Let res_s denote the resultant with respect to s. Then we compute two Sylvester resultants:

$$\begin{aligned} R_1(t, r) &= \mathrm{res}_s(Q(t, r, s), B(t, s)) \\ &= \underbrace{(at^{28}r^{24} + \ldots)}_{R_1'}(b_{12}t^{12} + \ldots + b_1 t + b_0)(1 + t^2)^4 \\ R_2(t) &= \mathrm{res}_r(R_1'(t, r), B(t, r)) \\ &= (a_{184}t^{184} + \ldots + a_1 t + a_0)(c_{12}t^{12} + \ldots + c_1 t + c_0)^6(1 + t^2)^{28}. \end{aligned}$$

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. We have proven that the factors exist, but we have no proof for the exponents.

We are now able to describe our subdivision scheme. We solve only for t and we show that if we know t with sufficient precision, then we are able to answer κ_3. The

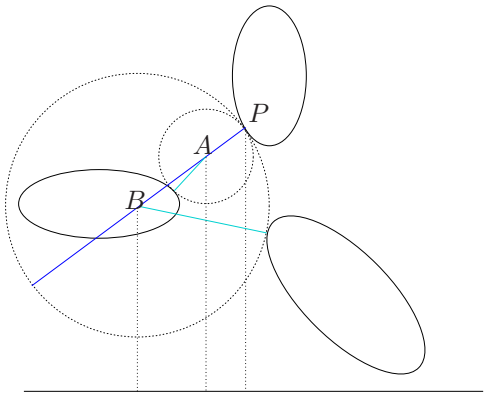

Fig. 10. Comparing radii f_{tr} and f_{ts}

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:

$$\mathcal{S}_{trs}(t) = \frac{P_t(t)(Q_1 - Q_2)(Q_1 + Q_2)}{4\big[(1+t^2)(1+r^2)(1+s^2)D_{tr}(t,r)D_{ts}(t,s)\big]^2}, \tag{16}$$

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

In the sequel we are interested in the sign of $\mathcal{S}_{trs}(t)$ at a given t. We observe that $Q = (Q_1 - Q_2)/(1 + w_1^2)$ which is the condition where three normals intersect at the same point.

Lemma 7. *To determine the sign of $\mathcal{S}_{trs}$ we only need to consider $Q(t, r, s)$, $A_{tr}(t, r)$ and $D_{ts}(t, s)$*

Proof. Consider the fig. 10, where $\overrightarrow{PA} \parallel \overrightarrow{PB}$ and the projections of the points P, A, B on x-axis (if the projections coincide, then we can consider the projections on the other axis). To compare $|\overrightarrow{PA}|$ and $|\overrightarrow{PB}|$ it suffices to compare the x-coordinates of P, A and B. More specifically,

$$\text{sign}(|\overrightarrow{PA}| - |\overrightarrow{PB}|) = \text{sign}((A_x - P_x)(A_x - B_x)).$$

The above expressions evaluate as follows:

$$A_x - P_x = \frac{y'(t)A_{tr}(t,r)}{(1+t^2)(1+r^2)D_{tr}(t,r)}$$
$$A_x - B_x = \frac{y'(t)Q(t,r,s)}{(1+t^2)(1+r^2)(1+s^2)D_{tr}(t,r)D_{ts}(t,s)},$$

where $y'(t)$ is the derivative of the y-coordinate of the parametric ellipse. □

We use a customized bisection to find a root of $\mathcal{S}_{trs}(t)$. Using the above lemma, we perform sign evaluations separately on A_{tr} and D_{ts}, instead of $Q_1 + Q_2$ because the former polynomials are of smaller degree.

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 operation is defined as follows:

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

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 $2^{-k}(b - a)$.

6.2. *Deciding κ_3*

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

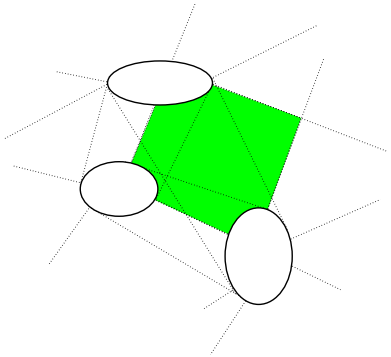

Fig. 11. Starting intervals for t, r, s (and region of the Voronoi vertex)

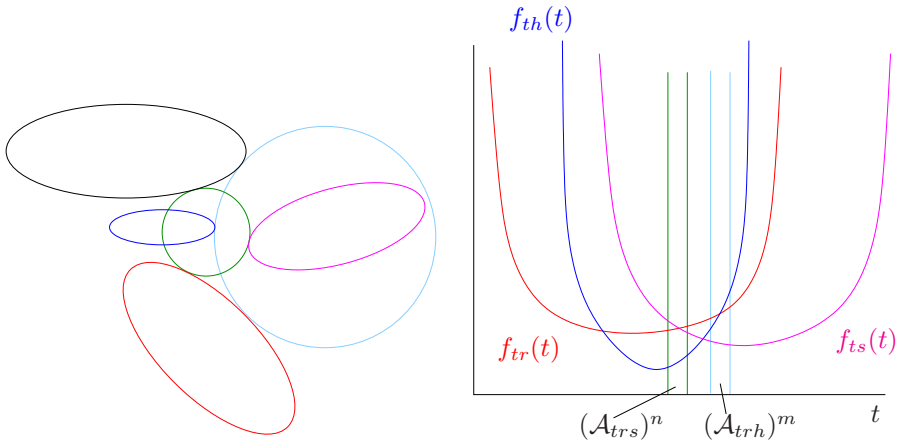

Fig. 12. Deciding κ_3

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 three. 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 8. *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 three 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 degenerate configuration. □

Observe that there is a neighborhood $\mathcal{U}$ of x where $\text{sign}(S_{trh}(u)) = \text{sign}(S_{trh}(x))$, $\forall u \in \mathcal{U}$. In our implementation, to find $\mathcal{U}$, it suffices that we have separated the roots of $\mathcal{S}_{trs}, \mathcal{S}_{trh}$. Fig. 12 shows an example where the query ellipse intersects the Voronoi circle.

We now establish the *exactness* of the subdivision algorithm, by computing the number of bits that suffice in order to certify the predicate. We shall use the system (11) that has optimal mixed volume. Recall that the system is

$$\Delta_1(v_1, v_2, q) = \Delta_2(v_1, v_2, q) = \Delta_3(v_1, v_2, q) = q - v_1^2 - v_2^2 + s = 0.$$

Let us eliminate v_1, v_2, q; the resultant $R(s)$ is of degree 184 in s and has coefficient bit size $3 \cdot 56 \cdot \tau_\Delta = 168\tau_\Delta$ [12]. 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 $1 \leq i, j \leq 3$ and $i \neq j$.

The minimum distance between two roots of a polynomial P (i.e. *separation bound*) of degree d and bit size τ is[32] $sep(P) \geq d^{-(d+2)/2}(d+1)^{(1-d)/2}2^{\tau(1-d)}$, 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 $|s_1 - s_2| \geq sep(R_1R_2)$, 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[32] 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 two.

This bound is close to tight, from a theoretical point of view, 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 worst case separation bound can be attained by Mignotte's polynomials[27]. To be more specific consider the (Mignotte) polynomial $x^d - 2(a\,x - 1)^2$, where $d \geq 3$ and $a \geq 3$, has two real roots in the interval $(\frac{1}{a} - h, \frac{1}{a} + h)$.

If the ellipses are given parametrically, in order to compute the implicit representation (3), the bit size increases by a factor of six. 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 $\log_\phi(1508 + 181944\tau)$.

6.3. *Overall method for κ_3*

The theoretical separation bound for the real roots of univariate polynomials that we presented in the previous section is a significant overestimation, in almost all

cases. For example the Wilkinson polynomial of degree 20, i.e $\prod_{i=1}^{20}(x-i)$, has a theoretical separation bound 10^{-344}, while the actual separation bound is 1. Even though we can not avoid it in the analysis, doing computations up to the theoretical separation bound turned out to be impractical. A small separation bound induces a big number of subdivision steps. For this we turn to an algebraic method.

Given three ellipses E_t, E_r, E_s and a query one E_h, we compute the resultants of system (15), i.e. $R(t)$ and $R'(t)$, with respect to the two triplets t, r, s and t, r, h, respectively. Recall that t is the subdivision parameter. If all four ellipses share a common Voronoi circle, then $R(t)$ and $R'(t)$ have a common root. In this case $G(t) = \gcd(R(t), R'(t)) \neq 1$. Given $R(t), R'(t)$ and $G(t)$, we can isolate the real roots of $R(t)$, $R'(t)$ and $G(t)$, and obtain the actual separation bound. Now, we may run the subdivision scheme up to this bound, which in all practical cases, is a lot larger than the theoretical one.

In practice we run the subdivision scheme for a predescribed number of steps, in order to decide almost all the easy cases. If we do not get the answer then we switch to the algebraic approach that we presented above. The real roots of $R(t)$, $R'(t)$ and $G(t)$ are represented by isolating intervals. If R and R' do not have common real roots, i.e $G(t) = 1$, then we continue the subdivision algorithm until it halts, since we know that the subdivision process stops. If they have common real roots, i.e $G(t) \neq 1$, then we refine the subdivision interval either until it contains only one real of $G(t)$, which means that we are in a degenerate configuration, or until it contains no real root of $G(t)$, which means that the subdivision process stops and answers the predicate.

Notice that in all the cases the subdivision scheme runs, in the worst case, up to the real separation bound induced by the real root isolation of $R(t)$ and $R'(t)$.

7. 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 six cases [14,22]: NoConflict, Interior, 123-Vertex, 124-Vertex, TwoVertices, EntireEdge (fig. 13).

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 four 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 three of the four ellipses and E_q. The possible outcomes are NoConflict, EntireEdge.

Let us now see the idividual outcomes of the predicate in more detail:

Cases 1 & 2: E_q is outside both Voronoi circles. It lies in the region between

case	$\kappa_3(E_t, E_r, E_s, E_q)$	$\kappa_3(E_t, E_r, E_h, E_q)$	κ_4
1	$0 \vee$ False	$0 \vee$ False	NoConflict
2			Interior
3	True	$0 \vee$ False	123-Vertex
4	$0 \vee$ False	True	124-Vertex
5	True	True	TwoVertices
6			EntireEdge

Table 1. Deciding κ_4

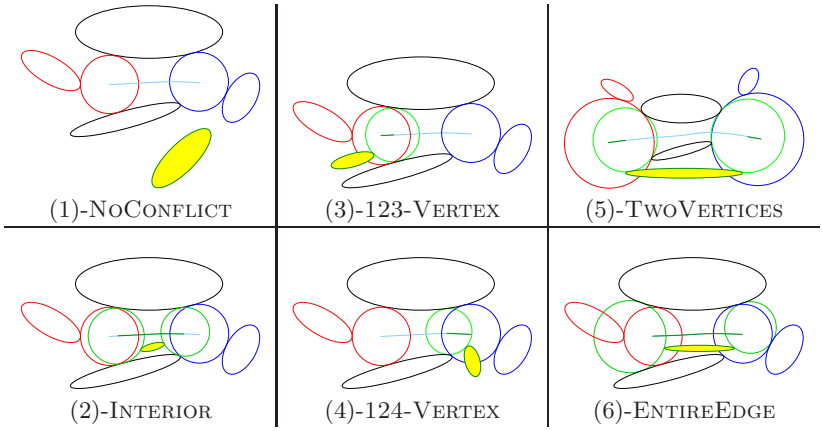

Fig. 13. All cases of κ_4, where the query ellipse (E_q)is shaded. Top, bottom, left and right ellipses are E_t, E_r, E_s and E_h respectively.

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 $[t_3, t_4]$, or in formal notation $(\mathcal{A}_{trq})^k \subseteq [t_3, t_4] \Longrightarrow$Interior. Otherwise, E_q does not conflict with the bisector, or more formally $(\mathcal{A}_{trq})^k \cap [t_3, t_4] = \emptyset \Longrightarrow$NoConflict.

Cases 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. In both cases, we have $(\mathcal{A}_{trq})^k \subseteq [t_3, t_4]$.

Cases 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, or equivalently $(\mathcal{A}_{trq})^k \subseteq [t_3, t_4] \Longrightarrow$TwoVertices. Otherwise, E_q conflicts with the entire edge, that is $(\mathcal{A}_{trq})^k \cap [t_3, t_4] = \emptyset \Longrightarrow$EntireEdge.

8. 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 [15,16], which are implemented in the SYNAPS library [28].

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 four we used the implementation based on the algorithms of [15], 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. 9.

For each test we randomly generated 1000 instances (point and two ellipses, or three 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 sub-quadratically 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 required. We suspect a degeneracy if after a certain number of iterations, our algorithm has not yet decided the predicate. Then we verify the degeneracy by computing the resultant, as described in section 6.3. This turned out to work very well

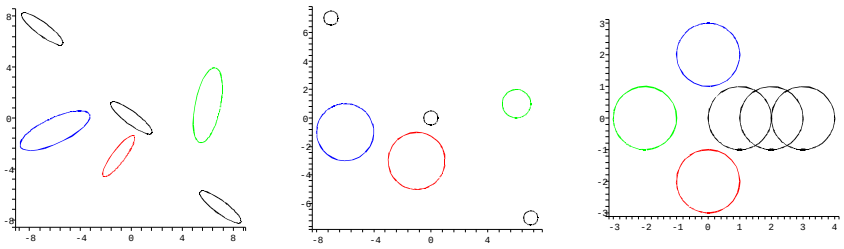

Fig. 14. Tested configurations of three ellipses (circles) and sample positions of the query object

in practice. With 10-bit coefficients, the computation of the resultant of (11) takes about five minutes with interpolation, while of (15) it is computed within seven seconds 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 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. 14 shows the three ellipses (or circles) and the query one in its initial, middle and final position. In fig. 15 we present the times for three 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 2.5 seconds. 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 [19]. Fig. 16 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.

9. 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 available.

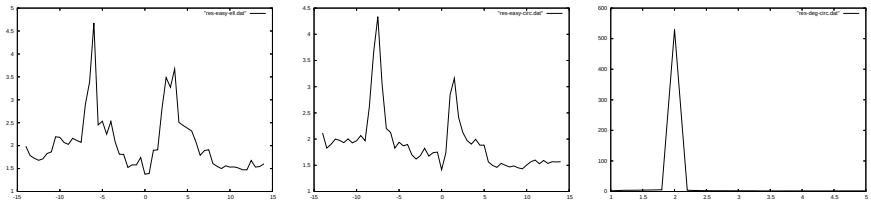

Fig. 15. Execution time of κ_3 as function of the position of the query ellipse's (circle's) center

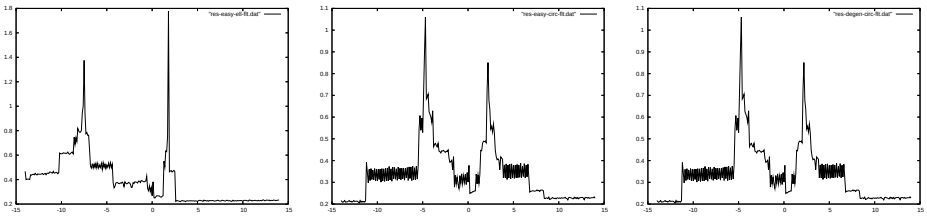

Fig. 16. Execution time of κ_3 as function of the position of the query ellipse's (circle's) center, using machine precision

Currently our subdivision scheme provides one bit per iteration, thus has a linear convergence. We could speed up the subdivision by using a method with better convergence rate. In general, for such methods to work the function (in our case $\mathcal{S}_{trs}$) should meet several criteria (like convexity or monotonicity). Even strict monotonicity alone would be an improvement, as we would be able to use Brent's (also known as Van Wijngaarden-Dekker-Brent's) method[10] with convergence rate $\phi = 1.618$ in general. The implementation of these methods should be based on multi-precision floats and interval arithmetic.

Last, but not least, we would like to certify our algorithm using the constructive root separation bounds[11,31], which may be tighter than the static bounds now used.

Acknowledgments

The authors would like to acknowledge S. Lazard and S. Petitjean at the 2nd ARCADIA Workshop for their useful comments and suggestions. We also thank S. Petitjean for contacting F. Ronga.

George Tzoumas is partially supported by State Scholarship Foundation of Greece, Grant No. 4631. 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. The project is co-funded by the European Social Fund & National Resources - EPEAEK II - PYTHAGORAS.

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. J.-D. Boissonnat and M. Teillaud, editors. *Effective Computational Geometry for Curves and Surfaces*. Springer-Verlag. ISBN: 3-540-33258-8. Not yet published. Available: January 3, 2007.
10. R. Brent. *Algorithms for Minimization without Derivatives*. Prentice-Hall, Englewood Cliffs, N.J., 1973.
11. 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.
12. D. Cox, J. Little, and D. O'Shea. *Using Algebraic Geometry*. Number 185 in GTM. Springer, New York, 2nd edition, 2005.
13. I. Z. Emiris, E. P. Tsigaridas, and G. M. Tzoumas. The predicates for the voronoi diagram of ellipses. In *Proc. ACM 22th Annual Symposium on Computational Ceometry (SoCG)*, pages 227–236, New York, NY, USA, 2006. ACM Press.
14. 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.
15. I.Z. Emiris and E.P. Tsigaridas. Computing with real algebraic numbers of small degree. In *Proc. ESA*, LNCS, pages 652–663. Springer, 2004.
16. 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.
17. 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*, 23(4):324–350, May 2006.
18. L. Habert. Computing bitangents for ellipses. In *Proc. 17th Canad. Conf. Comp. Geom.*, pages 294–297, 2005.
19. 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. (Best paper award).
20. 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.
21. F. Hartmann and R. Jantzen. Apollonius' ellipse and evolute revisited. Manuscript. http://www42.homepage.villanova.edu/frederick.hartmann/resume.html, September 2003.
22. M.I. Karavelas and M. Yvinec. Voronoi diagram of convex objects in the plane. In *Proc. ESA*, pages 337–348, 2003.
23. 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.
24. R. Klein, K. Mehlhorn, and S. Meiser. Randomised incremental construction of abstract Voronoi diagrams. *Comput. Geom. Theory & Appl.*, 3(3):157–184, 1993.
25. D. Lazard, 2006. Personal communication.
26. 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.
27. M. Mignotte. Some useful bounds. In B. Buchberger, G.E. Collins, and R. Loos, editors, *Computer Algebra: Symbolic and Algebraic Computation*, pages 259–263. Springer-Verlag, Wien, 2nd edition, 1982.
28. 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.
29. F. Ronga, 2005. Personal communication.
30. W. Wang, J. Wang, and M. Kim. An algebraic condition for the separation of two ellipsoids. *Comp. Aided Geom. Design*, 18:531–539, 2001.
31. 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.
32. C.K. Yap. *Fundamental Problems of Algorithmic Algebra*. Oxford University Press, New York, 2000.