An Efficient Algorithm for the InCircle Predicate among Smooth Closed Curves

Ioannis Z. Emiris* George M. Tzoumas†

Abstract

This paper concentrates on the InCircle predicate which is used for the computation of the Voronoi diagram of smooth closed curves. The predicate decides the position of a query object relative to the Voronoi circle of three given ones. We focus on (non-intersecting) ellipses but our method extends to arbitrary closed smooth curves, given in parametric representation. We describe an efficient algorithm for InCircle based on a certified numeric algorithm. The algorithm relies on a geometric preprocessing that guarantees a unique solution in a box of parametric space, where a customized subdivision-based method approximates the Voronoi circle tracing the bisectors. Our subdivision method achieves quadratic convergence by exploiting the geometric characteristics of the problem. The paper concludes with experiments showing that most instances run in less than 0.1 sec using floating-point arithmetic, on a 2.6GHz Pentium-4.

1 Introduction

The InCircle predicate is used in the incremental computation of the Voronoi diagram of curved objects [5]. Exact computation of this predicate is hard, since there may exist up to 184 (complex) tritangent circles to 3 conic curves; see [3] for a proof in the case of ellipses. In that work, the authors sketched a subdivision scheme for InCircle that worked better than generic solvers, although it had linear convergence. Notice that solving the algebraic system that defines the Voronoi circle is not enough to decide InCircle; one must take into account the query ellipse as well, which is equivalent to solving another similar system. In this approach, the computation of the Voronoi circle is the most computationally demanding task of InCircle.

The work coming closest to ours towards a complete Voronoi diagram is [4]. 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 and their software[1] works well in practice, with runtimes ranging from a few seconds to a few minutes.

In this paper, we focus on the computation of InCircle which is clearly the most challenging predicate and the only one that is not satisfactorily answered yet. Our main contribution is a new numeric algorithm that exhibits *quadratic* convergence in order to approximate the Voronoi circle, by exploiting several geometric properties of the problem. Not only is our method faster than generic solvers and other existing implementations, including the one in [3], but also allows us to decide InCircle before full precision has been achieved. An advantage of this method is that it can be generalized to arbitrary parametric curves and arcs. The algorithm proved to be very efficient, approximating the root with precision up to 10^{-15} in less than 0.1 sec, when using standard floating-point arithmetic. This work has been presented in a more complete context in [2].

We assume that an ellipse is given in rational parametric form, i.e. constructively, in terms of its rational axes, center and rotation:

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

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 center. We assume that all ellipses are parameterized in the same direction, i.e. CCW.

The *bisector* of two ellipses is the locus of points at equal distance from the two ellipses. It can be shown that the bisector of two ellipses in the parametric space, is a bivariate polynomial $B_1(t, r)$ of total degree 12, six in each variable. Each point on the bisector corresponds to the center of a circle which is bitangent to the two ellipses. For InCircle we are interested in that part of the bisector which is the locus of centers of externally bitangent circles.

Lemma 1 *Given two smooth closed curves and a point on the first, there is a bounded number of real bitangent circles, tangent at the specific point. This number is 6 for conics and is tight for ellipses.*

[*] Department of Informatics and Telecoms, National University of Athens

[†] Department of Informatics and Telecoms, National University of Athens, geotz@di.uoa.gr

[1] www.cs.technion.ac.il/~irit/

The proof of the above lemma (for ellipses) as well as more details on bisectors of ellipses can be found in [3]. For parametric bisectors, the reader may also refer to [1]. The smoothness property implies that a unique normal line is defined at every point of the curve.

Note that from the circles in the previous lemma, only one is *external* to both ellipses. We call this unique external bitangent circle the *Apollonius* circle of the two ellipses. We denote the Apollonius circle of E_t and E_r tangent at points $\hat{t}$ and $\hat{r}$ respectively by $\mathcal{A}_{tr}(\hat{t}, \hat{r})$. Since $\hat{r}$ depends on $\hat{t}$, we may omit the latter and write only $\mathcal{A}_{tr}(\hat{t})$.

In the parametric space, the intersection of two bisectors involves three variables. In order to express the Voronoi circle, we consider the intersection of three bisectors by solving the system:

$$B_1(t, r) = B_2(s, t) = B_3(r, s) = 0. \qquad (2)$$

2 Voronoi circle using subdivision

We have tried the ALIAS[2] [6] built-in subdivision solvers. The naive subdivision algorithm failed to converge, while gradient based methods converged within a few seconds depending on the initial interval.

Here we present a subdivision scheme to approximate the solution of system (2) which is simpler, exhibits quadratic convergence, and exploits the system's geometric symmetry.

Lemma 2 *Consider $\mathcal{A}_{tr}(t, r)$, which means that $B_1(t, r) = 0$. If we slide this circle along the boundary of the ellipses for a sufficiently small amount, while varying its radius and keeping it tangent to t' and r' respectively, then $t' > t \Longrightarrow r' < r$.*

Proof. This lemma can be proved by contradiction. See figure 2 for a sketch of the proof. □

The basic idea of our algorithm is the following: Let $(\hat{t}, \hat{r}, \hat{s})$ be the solution of (2) we are looking for. Now consider the following system:

$$\begin{aligned} B_1(t_1, r_2) = B_3(r_2, s_1) = B_2(s_1, t_2) &= 0 \qquad (3) \\ B_1(t_2, r_1) = B_3(r_1, s_2) = B_2(s_2, t_1) &= 0 \qquad (4) \end{aligned}$$

These two systems look like (2). The difference is that we have considered $t_1 \neq t_2$ in the general case and thus we can start solving the above equations in the given order. Doing so and keeping solutions that correspond to Apollonius circles (using the geometric arguments described in [3]), leads to a construction as in fig. 1. All bitangent circles coincide with the Voronoi circle when $t_1 = \hat{t} = t_2$. Otherwise, we have found an interval $[t_1, t_2]$ that contains $\hat{t}$. We can refine it by choosing a new point t_1' inside this interval and computing $[t_1', t_2']$ (suppose without any harm that $t_1' < t_2'$). As a consequence of lemma 2, t_1 approaches $\hat{t}$ from the left ⇒ r_2 approaches $\hat{r}$ from the right ⇒ s_1 approaches $\hat{s}$ from the left ⇒ t_2 approaches $\hat{t}$ from the right (see fig. 1). Therefore $t_1' \in [t_1, t_2] \Longrightarrow \hat{t} \in [t_1', t_2'] \subset [t_1, t_2]$. Note that computing a smaller interval on dimension t allows us to compute smaller intervals on dimensions r and s as well. Therefore we maintain exactly one box that contains our solution, contrary to generic interval-arithmetic techniques that may need to maintain a large number of boxes.

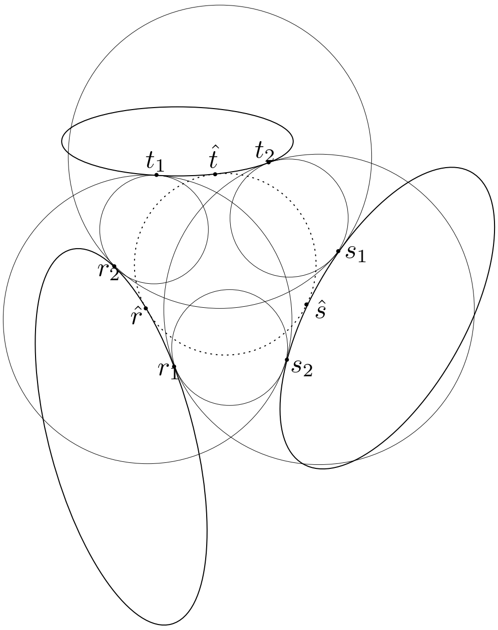

Figure 1: All-pair bitangent circles

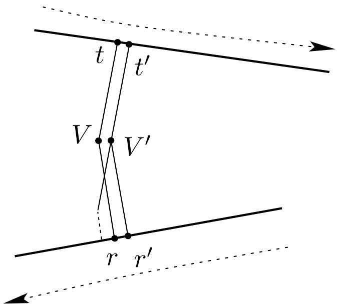

Figure 2: $t' > t \Longrightarrow r' < r$

2.1 Starting interval

First, we show that it is possible to find a box that contains a unique solution. Consider the half-planes l_1, l_2 which do not contain the ellipses and are bounded by the two external bitangents of E_1 and E_2.

[2] www-sop.inria.fr/coprin/logiciels/ALIAS/ALIAS.html

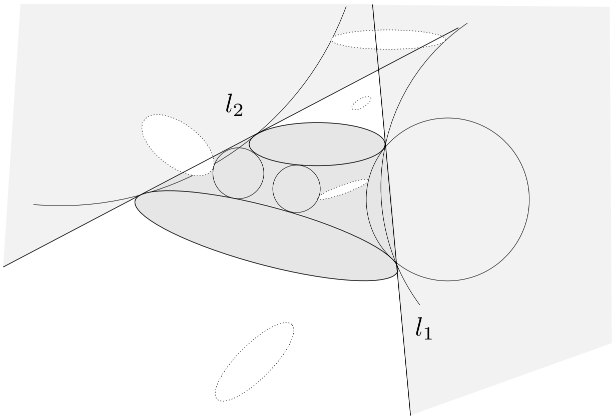

Figure 3: The cases on the number of Voronoi circles, depending on the position of the third dotted ellipse

Consider also a query ellipse E that does not intersect the other two. (cf. fig. 3) Let $|l_i| = 0$ or 1 depending on whether $E \cap l_i = \emptyset$ or not. Let C be the interior of the convex hull of E_1, E_2. Then, the number of Voronoi circles is 0, 1 or 2, namely $|l_1| + |l_2|$, if $E \cap C = \emptyset$ and $2 - |l_1| - |l_2|$ otherwise.

We can find a starting interval that contains the tangency point of the Voronoi circle by computing the external bitangents of each pair of ellipses and taking the intersection of the interior of their convex hulls. After computing the initial interval, we can pick any random point t_1 to start our algorithm.

These intervals' endpoints may not correspond to the same bitangent circle. Then, it is necessary to "normalize" them so that they contain some solution of (3) and (4), cf. fig. 1. We start from value t_1. At any given step, we have computed a value for parameter t, say $\bar{t}$ and then compute the Apollonius circle at point $\bar{r}$ of the next ellipse, that is we compute $\mathcal{A}_{tr}(\bar{t}, \bar{r})$ (by solving $B_1(\bar{t}, r)$ with respect to r). If $\bar{r} \in [r_1, r_2]$, then we update one appropriate endpoint to $\bar{r}$. If $\bar{r} \notin [r_1, r_2]$, then we update $\bar{t}$ by computing the Apollonius circle tangent to the corresponding endpoint of $[r_1, r_2]$. This process is analogous for the other parameter pairs (r, s) and (s, t).

There is also the case where two Voronoi circles exist. In this case the overall algorithm indicates which one is needed and therefore we pick a proper subinterval.

2.2 Subdivision

The subdivision algorithm in pseudo-code is as follows:

Subdivision Algorithm

∗ INPUT: Initial intervals $[t_1, t_2], [r_1, r_2], [s_1, s_2]$ that contain unique $(\hat{t}, \hat{r}, \hat{s})$ and $\sigma \in \mathbb{R}$, $\sigma > 0$.

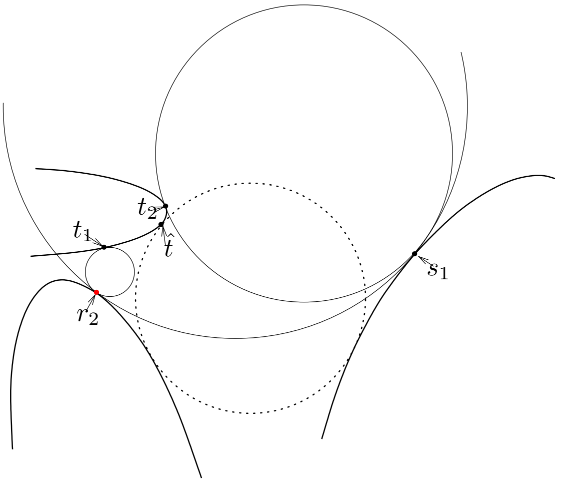

Figure 4: Computing t_2 from t_1 encloses $\hat{t}$

∗ OUTPUT: Subintervals $[t_1, t_2], [r_1, r_2], [s_1, s_2]$ of the given ones, which contain $(\hat{t}, \hat{r}, \hat{s})$ and $t_2 - t_1 < \sigma$.

1. Start from point t_1 on the first ellipse and solve system (3), with the additional constraint that r_2, s_1, t_2 correspond to Apollonius circles. After computing t_2, also solve (4) in order to obtain intervals for $[r_1, r_2]$ and $[s_1, s_2]$ respectively. For each interval, set the left endpoint to be smaller than the right one, by swapping them if necessary.
2. If $t_2 - t_1 < \sigma$ then stop.
3. Set $t_1 := \frac{t_1+t_2}{2}$. Note that the midpoint of $[t_1, t_2]$ could be on the left or on the right of $\hat{t}$. Go to step 1.

Eliminating r_2, s_1 from (3) yields resultant $R(t_1, t_2)$ as a bivariate polynomial equation. Similarly, eliminating r_1, s_2 from (4) yields $R(t_2, t_1)$ as a bivariate polynomial equation. Now we observe that $R(t_1, t_2) = R(t_2, t_1)$, since they have been derived from the same equations with the same coefficients. When $t_1 \neq t_2$, systems (3) and (4) express a family of circles bitangent to each pair of ellipses, as in fig. 1.

Looking at $R(t_1, t_2)$ we see that t_2 is an implicit function of t_1, say f, that is $f(t_1) = t_2$. Given value t_1, $f(t_1)$ is the value of t_2 after solving (3), shown in fig. 4. Obviously $f(\hat{t}) = \hat{t}$. From lemma 2 it follows that $t_1 < t' < \hat{t} \Longrightarrow f(t_1) > f(t') > f(\hat{t})$. That is, as t_1 approaches $\hat{t}$ from the left, t_2 approaches $\hat{t}$ from the right and $\hat{t} \in [t_1, t_2]$.

Lemma 3 *In the above notation $f'(\hat{t}) = -1$.*

Proof. From the Implicit Function Theorem we have that $\frac{df}{dt_1}$ exists and (by chain rule): $\frac{df}{dt_1} = f'(t_1) = -\frac{\partial R/\partial t_1}{\partial R/\partial t_2}$ where R is the resultant polynomial. Since $R(t_1, t_2) = R(t_2, t_1)$, at point $t_1 = t_2 = \hat{t}$ we have $\left.\frac{\partial R(t_1,t_2)}{\partial t_1}\right|_{t_1=t_2=\hat{t}} = \left.\frac{\partial R(t_1,t_2)}{\partial t_2}\right|_{t_1=t_2=\hat{t}}$, therefore $f'(\hat{t}) = -\left.\frac{\partial R/\partial t_1}{\partial R/\partial t_2}\right|_{t_1=t_2=\hat{t}} = -1$. □

Theorem 4 (Convergence) *The above subdivision algorithm converges quadratically.*

Proof. For the proof, t_1, t_2 are not generic variables, but have specific values, as is the case during the execution of the algorithm. Let

$$\epsilon = |f(t_1) - t_1|$$

be the error at one iteration of the method. At the next iteration, the new error is $\epsilon' = |f(\frac{t_1+f(t_1)}{2}) - \frac{t_1+f(t_1)}{2}|$ or equivalently: $\epsilon' = |f(t_1 + \frac{f(t_1)-t_1}{2}) - t_1 - \frac{f(t_1)-t_1}{2}|$. Applying Taylor expansion around point t_1 for $f(t_1 + \frac{f(t_1)-t_1}{2})$ yields $\epsilon' = |f(t_1) + \frac{f(t_1)-t_1}{2} f'(t_1) + \frac{(f(t_1)-t_1)^2}{8} f''(\xi) - t_1 - \frac{f(t_1)-t_1}{2}|$, with ξ between t_1 and $\frac{t_1+t_2}{2}$. Notice that, from the theory of Taylor expansion, the use of ξ allows us to omit less significant terms. Now ϵ' becomes: $\epsilon' = \left|\frac{f(t_1)-t_1}{2}(1 + f'(t_1)) + \frac{\epsilon^2}{8} f''(\xi)\right|$. This time we combine the Taylor expansion around point $\hat{t}$ for $f'(t_1)$ with lemma 3: $f'(t_1) = f'(\hat{t}) + (t_1 - \hat{t}) f''(\xi') = -1 + (t_1 - \hat{t}) f''(\xi')$, with ξ' between t_1 and $\hat{t}$, such that it allows us to omit less significant terms. Now remember that $\hat{t} \in [t_1, t_2]$ which means that $|t_1 - \hat{t}| \leq |t_2 - t_1| = |f(t_1) - t_1|$. Therefore: $\epsilon' = \left|\frac{f(t_1)-t_1}{2}(t_1 - \hat{t}) f''(\xi') + \frac{\epsilon^2}{8} f''(\xi)\right| \Longrightarrow$

$$\epsilon' \leq \frac{\epsilon^2}{2} \left|f''(\xi')\right| + \frac{\epsilon^2}{8} \left|f''(\xi)\right|.$$

Given that $f''(t)$ is a continuous function it takes a minimum and maximum value inside $[t_1, t_2]$, therefore $|f''(\xi)|$ and $|f''(\xi')|$ are bounded by a positive constant C and eventually $\epsilon' \leq \frac{5C}{8}\epsilon^2$. □

3 Conclusion

We have implemented the subdivision algorithm in C++ using the interval-arithmetic library ALIAS, therefore the results are certified up to floating-point precision. At each iteration we have to solve the bisector polynomial $B_1(t, r)$ for a fixed t. Fortunately ALIAS provides a fast univariate polynomial solver for this task. Computing the diameter of the approximating intervals during each iteration of the algorithm verified its quadratic convergence. For instance, a diameter sequence was: [3.01679, 0.978522, 0.303665, 0.0381727, 0.000628676, 1.70776e-07, 1.17684e-14]. Our experiments show that about 8 iterations are enough to approximate the roots of the system with a precision of 10^{-15} in about 80 msec on a Pentium-4 2.6GHz.

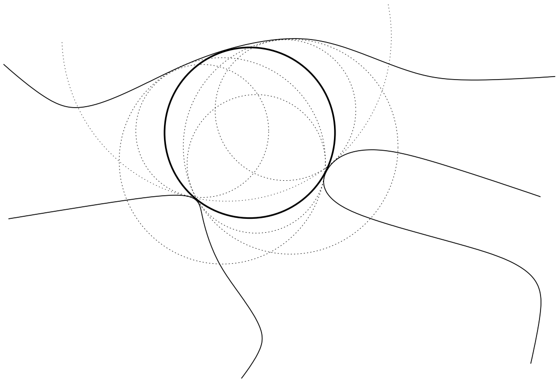

Figure 5: Voronoi circle of three arbitrary curves

The presented algorithm can be readily generalized in order to compute the Voronoi circle of arbitrary parametric curves, as shown in fig. 5. A subdivision algorithm alone does not suffice to decide the predicate in degenerate cases. To handle those cases, we apply resultants and real solving [3]. Along these lines, we plan to develop C++ code that will lead to a complete CGAL[3] package.

Acknowledgments

George Tzoumas is partially supported by State Scholarship Foundation of Greece, Grant No. 4631. Both authors acknowledge 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).

References

[1] G. Elbert and M.-S. Kim. Bisector curves of planar rational curves. *Computer-Aided Design*, 30:1089–1096, 1998.

[2] I. Emiris and G. Tzoumas. A real-time and exact implementation of the predicates for the Voronoi diagram of parametric ellipses. Submitted. Manuscript available from http://www.di.uoa.gr/~geotz/.

[3] 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, Sedona, Arizona, USA, 2006. ACM Press.

[4] 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).

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

[6] J.-P. Merlet. ALIAS: an interval analysis based library for solving and analyzing system of equations. In *Systèmes d'Équations Algébriques*, Toulouse, France, 2000.

[3] www.cgal.org