Exact Delaunay graph of smooth convex pseudo-circles

Ioannis Z. Emiris* Elias P. Tsigaridas† George M. Tzoumas*

Abstract

We examine the problem of computing exactly the Delaunay graph of a set of possibly intersecting smooth convex pseudo-circles in the Euclidean plane given in parametric form. The diagram is constructed incrementally. We focus on InCircle, under the exact computation paradigm, and express it by a simple polynomial system, which allows for an efficient implementation by means of iterated resultants and a factorization lemma. Finally, we present examples with various types of curves.

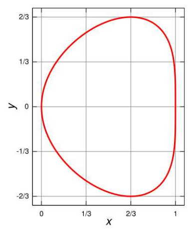

Fig. 1.1: The Bean curve $t \mapsto (\frac{1+t^2}{t^4+t^2+1}, \frac{t(1+t^2)}{t^4+t^2+1})$.

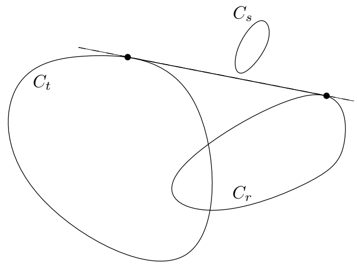

Fig. 1.2: Demonstration of DistanceFromBitangent.

1 Introduction

Computing the Delaunay graph and its dual Voronoi diagram of a set of input sites in the plane has been studied extensively due to its numerous applications. However, few works have studied *exact* Voronoi diagrams[1] for curved objects. These can be critical in applications such as assembly, surface reconstruction, and crystallography. In the case of circles, the exact and efficient implementation of [3] is now part of CGAL [2]. There is also VRONI, a very efficient and robust implementation, which relies on floating-point computations; a more recent implementation of which can treat (non intersecting) line segments and circular arcs [8]. Conics were considered in [1], but only in a purely theoretical framework. Moreover, the algebraic conditions derived were not optimal, leading to a prohibitively high algebraic complexity. In fact, the approach relied on eigenvector computations, hence was not exact.

Our own previous work [5] started with the study of non-intersecting ellipses, and proposed exact algebraic algorithms for all predicates required by the incremental algorithm of [9]. The resultant required had been implemented in MAPLE, and used a different polynomial system than the one in this paper. Moreover, some factorization properties had been observed without proof; this is settled below to yield an optimal method for resultant computation. In [6], the authors proposed a certified method for InCircle, relying on a Newton-like numerical subdivision, which exploits the geometry of the problem and exhibits quadratic convergence for non-intersecting ellipses.

In this paper, we extend previous results that considered only non-intersecting ellipses. We study the case of smooth convex, possibly intersecting, pseudo-circles. We first model the definition of conflict from [9] as a circle inclusion test. Then, we examine the algebraic operations required for an efficient exact implementation of InCircle which shows high algebraic complexity dominating the overall algorithm.

Notation and preliminaries. Our input is smooth convex closed curves given in parametric form. An example of such a curve, the famous *bean* curve is in Fig 1.1. Smoothness allows the tangent (and normal) line at any point of the curve to be well-defined. We denote by C_t a smooth closed convex curve parametrized by t. We refer to a point p on C_t with parameter value $\hat{t}$ by $p_{\hat{t}}$, or simply by $\hat{t}$ when it is clear from context. By ${C_t}^\circ$ we denote the interior of curve C_t. $\mathbf{C_t}$ is a smooth convex object (site), so that if p denotes a point in the plane, $p \in \mathbf{C_t} \iff p \in C_t \cup {C_t}^\circ$. When we say that two sites intersect, we assume that their boundaries have at most two intersections, i.e. they form *pseudo-circles*. A curve C_t is given by the map

$$C_t : [a,b] \ni t \mapsto (X_t(t), Y_t(t)) = (F_t(t)/H_t(t), G_t(t)/H_t(t)), \quad (1)$$

where F_t, G_t and H_t are polynomials in $\mathbb{Z}[t]$, with degrees bounded by d, and $a, b \in \mathbb{Q} \cup \{\pm\infty\}$. All algorithms, predicates and the corresponding analysis are valid for any parametric curve, even when the polynomials have different degrees, including the case of different denominators. We assume equa-

*National Kapodistrian University of Athens, Hellas. `{emiris, geotz} (AT) di.uoa.gr`

†INRIA Méditerranée, Sophia-Antipolis, France, `elias.tsigaridas (AT) sophia.inria.fr`

[1] The Voronoi diagram cannot be represented "exactly", since it involves algebraic numbers, although our methods allow for an arbitrary precision approximation of such numbers, and this precision need not be fixed in advance. On the other hand, the dual Delaunay graph is represented exactly.

tions (1) for simplicity. Moreover, we assume that $H_t(t) \neq 0$, $t \in [a, b]$. For simplicity we write F_t instead of $F_t(t)$ and denote its derivative with respect to t as F_t'. When $d = 2$ the curves defined are conics: ellipses and circles are the only closed convex curves represented.

Basic predicates. The insertion of a new site in the current Voronoi diagram consists of the following: (i) Find a conflict between an edge of the current diagram and the new site, or detect that the latter is internal (hidden) in another site, in which case it does not affect the diagram. (ii) Find the entire *conflict region* (the part of the Voronoi diagram that changes due to the insertion of the new site) and update the dual Delaunay graph. It should be noted here that our main task is to compute an exact Delaunay graph of the input sites (which maintains exact topology information). Having computed an exact Delaunay graph, allows us to approximate the vertices (bisectors) and edges of the Voronoi diagram (the coordinates of which are algebraic numbers) within an arbitrary precision in order to be drawn on the screen. The exact graph allows us to mark correctly the degenerate cases, i.e., Voronoi vertices of degree > 3.

The above algorithm requires the following predicates. (a) SideOfBisector: given two sites $\mathbf{C_t}$ and $\mathbf{C_r}$ and point q, determine the site closest to the point, under the Euclidean metric, (b) DistanceFromBitangent: given two sites, $\mathbf{C}_t$ and $\mathbf{C}_r$, decide the position of a third site, $\mathbf{C}_s$, with respect to the external bitangent line of the first two, that leaves both sites on the right, as we move from the tangency point of $\mathbf{C}_t$ to the tangency point of $\mathbf{C}_r$. The result of DistanceFromBitangent is either 1 if $\mathbf{C}_s$ lies on the same hyperplane as the sites $\mathbf{C}_r, \mathbf{C}_r$ w.r.t. their external bitangent, 0 if it is tangent to that line, but on the same hyperplane, and -1 otherwise (cf. fig. 1.2). (c) InCircle, and (d) EdgeConflictType. Some operations require two additional primitives: (i) computing a rational point inside the convex site and (ii) determining the position between two sites, i.e., whether they are separated. A detailed presentation of SideOfBisector, DistanceFromBitangent, EdgeConflictType and the primitives is omitted due to space restrictions, but can be found in [4]. Predicate InCircle is presented in the next section.

Normal line. A point on the curve is denoted by $p_t = (X_t, Y_t)$. The equation of the line that supports the *normal* at p_t is $(N_t) : (x - X_t)X_t' + (y - Y_t)Y_t' = 0$. After substitutions and elimination of the denominators, we derive a polynomial $N_t(x, y, t) \in \mathbb{Z}[x, y, t]$; which is linear in x and y, of degree $\leq 3d - 2$ in t.

2 InCircle

We first formalize the notion of conflict, caused by the addition of a new site [9], (see also fig. 2.1 and 2.2):

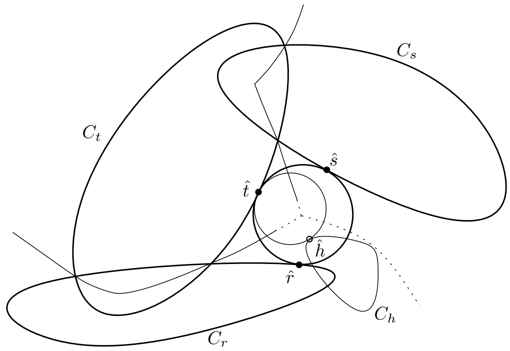

Fig. 2.1: Lem. 2: conflict of query site $\mathbf{C_h}$ with external Voronoi disk.

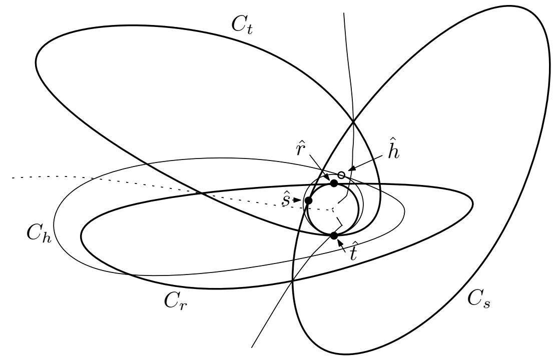

Fig. 2.2: Lem. 3: conflict of $\mathbf{C_h}$ with internal Voronoi disk; the Voronoi edges are solid, the conflict region (edge) is dotted.

Definition 1 *Given sites $\mathbf{C_t}$, $\mathbf{C_r}$, $\mathbf{C_s}$, let V_{trs} be their Voronoi disk and $\mathbf{C_h}$ be a query site. If V_{trs} is an external Voronoi disk, then $\mathbf{C_h}$ is in conflict with V_{trs}, iff V_{trs} is intersecting $C_h{}^\circ$. If V_{trs} is an internal Voronoi disk, then $\mathbf{C_h}$ is in conflict with V_{trs}, iff V_{trs} is included $C_h{}^\circ$.*

Since the Voronoi circle in expressed algebraically, we cannot decide its relative position w.r.t. the query site by counting bitangent lines. Therefore, we model the above disk inclusion test as a circle inclusion test.

Lemma 2 *Given convex sites $\mathbf{C_t}$, $\mathbf{C_r}$, $\mathbf{C_s}$, let V_{trs} be an external Voronoi disk of theirs and $\hat{t}$ its tangency point on C_t. Let $\mathbf{C_h}$ be a query site and B_{th} an external bitangent disk of $\mathbf{C_t}$ and $\mathbf{C_h}$, tangent at $\hat{t}$ (and $\hat{h}$ resp.). Then $\mathbf{C_h}$ is in conflict with V_{trs} if and only if B_{th} is strictly contained in V_{trs}.*

Now it remains to compute the external bitangent circle B_{th}. There may exist up to 6 bitangent circles, tangent at a given $\hat{t}$, for the case of ellipses [5], and up to a constant number for arbitrary convex sites. In [5], simple geometric tests are provided to isolate the external bitangent circle among all bitangent circles to non-intersecting ellipses. Now, consider two intersecting sites $\mathbf{C_t}$ and $\mathbf{C_r}$ and an external bitangent circle tangent at point $\hat{t}$ of C_t, then we have: (i)

$\hat{t} \notin C_t \cap C_r$ since it is the tangency point of an externally tangent circle, (ii) $\hat{t}$ lies on the arc of C_t bounded by the convex hull of $\mathbf{C_t}$ and $\mathbf{C_r}$ (iii) the tangent line of C_t at $\hat{t}$ intersects C_r.

The above conditions have been used to compute an external bitangent circle in the case of non-intersecting ellipses [5] when the tangent line at the given tangency point intersects the other ellipse, which is always the case when the two sites intersect. Therefore the geometric tests of [5] can be also applied to pseudo-circles. These tests yield an arc on the second site C_r that contains the tangency point of the externally tangent circle.

Intersecting sites may also admit an internally tritangent Voronoi circle. In this case InCircle can be answered by the following lemma:

Lemma 3 *Given sites $\mathbf{C_t}$, $\mathbf{C_r}$, $\mathbf{C_s}$, let V_{trs} be their internal Voronoi disk, and $\hat{t}$ its tangency point on C_t. Let $\mathbf{C_h}$ be a query site and B_{th} an internal bitangent disk of $\mathbf{C_t}$ and $\mathbf{C_h}$, tangent at $\hat{t}$ (and $\hat{h}$ resp.). Then $\mathbf{C_h}$ is in conflict with V_{trs} if and only if V_{trs} is strictly contained in B_{th}.*

Note that if $\hat{t} \notin \mathbf{C_h}$, then $\mathbf{C_h}$ is not in conflict with V_{trs} which can be used as an additional test in the implementation to quickly answer some cases.

Let us now identify an internal bitangent circle (among all bitangent circles). First, we observe that if an internal bitangent circle at point $\hat{t}$ of C_t exists, then $\hat{t} \in \mathbf{C_t} \cap \mathbf{C_r}$ and $\hat{r} \in \mathbf{C_t} \cap \mathbf{C_r}$ (the converse may not be necessarily true).

Lemma 4 *Given intersecting sites $\mathbf{C_t}$ and $\mathbf{C_r}$, consider their bitangent circle B_{tr} at points $\hat{t}$ and $\hat{r}$ respectively with $\hat{t}, \hat{r} \in \mathbf{C_t} \cap \mathbf{C_r}$. Then B_{tr} is an internal bitangent circle if and only if B_{tr} has the smallest radius among all bitangent circles of C_t and C_r tangent at $\hat{t}$, and the radius of B_{tr} is bounded by the radius of curvature of C_t at $\hat{t}$ and the radius of the self-bitangent circle of C_t at $\hat{t}$.*

In an implementation, the curvature constraint can be forced by comparing with the evolute point [7]. The self-bitangent circles can be computed by considering the bisector of a curve and itself, then by computing the (self-)bitangent circles and finally choosing the one with the smallest radius (in case of identical self-bitangent circle we can consider the evolute point).

Expressing the Voronoi circle. First, we consider the question of choosing, among all solutions of the polynomial system, the one corresponding to the Voronoi circle. The polynomial system expressing all circles tangent to $\mathbf{C_t}$, $\mathbf{C_r}$, $\mathbf{C_s}$ is:

$$\begin{aligned} N_t(x,y,t) = N_r(x,y,r) = N_s(x,y,s) &= 0 \\ M_{tr}(x,y,t,r) = M_{ts}(x,y,t,s) &= 0. \end{aligned} \tag{2}$$

The first 3 equations correspond to normals at points t, r, s on the 3 given sites. All normals go through the Voronoi vertex (x, y). The last two equations force (x, y) to be equidistant from the sites: each corresponds to the bisector of the segment between two footpoints. This system was also used in [10]. Notice that elimination of x, y from M_{tr}, N_t, N_r yields the bisector of two sites with respect to t, r.

A Voronoi circle is either externally or internally tritangent and its tangency points on C_t, C_r, C_s respectively have a CW or CCW orientation. Given a CCW orientation of the sites, either (t, r, s) or (t, s, r), we wish to identify the solution of the system that corresponds to that circle. First, we check if such a Voronoi circle exists. This is a generalization of the Existence sub-predicate of [3] to pseudo-circles. It can be performed for an external Voronoi circle without solving system (2) as follows: Sites C_t, C_r, C_s admit an external CCW Voronoi circle, iff there are at least two negative results of DistanceFromBitangent evaluated at triplets (C_t, C_r, C_s), (C_r, C_s, C_t) and (C_s, C_t, C_r) (in this cyclic order). This can be proven by enumeration, or by studying the adjacency relationship with the site at infinity. Checking that C_t, C_r, C_s admit an internal CCW Voronoi circle is more complex: First, we compute their intersection, which must be nonempty. Then, their intersection must have a CCW sequence of arcs on its boundary. Finally, we have to verify that system (2) has a solution corresponding to the Voronoi circle.

Solving system (2) over the reals, yields a set of solution vectors in $\mathbb{R}^5$. Only one solution vector contains the Voronoi vertex and the corresponding tangency points. There exist solution vectors with CW orientation, but also with CCW orientation which do not correspond to the Voronoi circle we are looking for, but to some other tritangent circle. At this point, we already know that an external Voronoi circle (with CCW orientation) exists, or that an internal Voronoi circle might exist. To eliminate irrelevant solutions, consider the tangency points $p_{\hat{t}}$, $p_{\hat{r}}$, $p_{\hat{s}}$ for a solution triplet $\hat{t}, \hat{r}, \hat{s}$. The tangency points corresponding to the Voronoi circle satisfy $\mathrm{CCW}(p_{\hat{t}}, p_{\hat{r}}, p_{\hat{s}})$.

Now we distinguish an external and an internal tritangent circle from the rest of the tritangent circles. The tangency points define the former iff the tangent line of the Voronoi circle at each tangency point separates its adjacent site from the other two tangency points. Even if the tangent line intersects the other sites (not shown in the example), the tangency points are still separated. Checking that the tangency points correspond to an internal circle can be performed by applying lemma 4. Finally, before proceeding with algebraic analysis, it has to be noted that in an implementation, the certified algorithm of [6] can be adapted in order to speed up the operations.

We examine system (2) so as to derive at an efficient algorithm for its solution. In general, the *resultant* of $n+1$ polynomials in n variables is an irreducible[2] polynomial in the coefficients of the polynomials which vanishes precisely when the system has a complex solution. Since it is impossible to compute the resultant of 5 general polynomials as a determinant, we compute it by successive Sylvester determinants, which are optimal formulae for the resultant when $n = 1$. This method typically produces extraneous factors but, by exploiting the fact that some of the polynomials are linear, and that none contains all variables, we shall be able to predict all such factors. The following theorem, the complete proof of which can be found in [4], provides the complete factorization of the resultant, in the case of conics.

Theorem 5 *We assume the resultant of (2) is nonzero and denote it by* Π. *Then,* $\mathsf{Res}_{xy}(R_1, R_2, N_t) = \Pi(t)H_t^{40}(G_tH_t' - G_t'H_t)^{36}$, *where,* $R_1 = \mathsf{Res}_r(M_{tr}, N_r)$, $R_2 = \mathsf{Res}_s(M_{ts}, N_s)$, *and* Π *is of degree 184.*

Corollary 6 *We are given* $R_0, R_1, R_2 \in \mathbb{K}[x, y]$, *where the total degree of* R_1 *and* R_2 *is* n *in* x, *in* y, *and in* x *and* y *together, and* $R_0 = Dy+Ax+C$, *where* $AD \neq 0$, *then* $\mathsf{Res}_x(\mathsf{Res}_y(R_0, R_1), \mathsf{Res}_y(R_0, R_2)) = D^{n^2}\mathsf{Res}_{xy}(R_0, R_1, R_2)$.

It follows that the degree of the resultant of (2) for general parametric curves, as in (1), is bounded by $(3d-2)(5d-2)(9d-2)$, after dividing out the factor of $(H_t(G_tH_t' - G_t'H_t))^{(5d-2)^2}$. A more careful analysis may exploit cancellations to yield a tighter bound.

3 Conclusion

We conclude this paper by applying the proposed algorithms for various types of curves.

Bean curve. We took the bean curve of fig. 1.1 and applied simple affine transformations yielding very small (5-bit) coefficients. Then, we computed the resultant of such a triplet. In this case $d = 4$, and the final resultant (after the factorization) was computed in Maple after 570 sec on a P4 2.6GHz and was found to be of degree 2632. The (non-tight) upper bound provided above yields 6120. The long runtimes indicate that working with high-degree curves might be non-practical.

Conics. In the case of conics, $d = 2$. We consider three random curves with small (10-bit) coefficients. Computing the final resultant of degree 184 (after the factorization) was possible in 8.5 sec on the same machine. Note that for $d = 2$ we have a tight bound, while the non-tight formula yields 512.

[2] Irreducibility occurs for generic coefficients. Below, certain resultants are factorized, because the given polynomials do not have generic coefficients.

B-splines. As a future work we may extend this approach to working with piecewise smooth polynomial curves. The critical part is to determine the "piece" of the function to apply the algebraic predicate. The tangency points of the Voronoi circle lie on such polynomial pieces, therefore the resultant formulation is completely analogous but simpler because of the lack of denominators.

References

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

[2] CGAL: Computational geometry algorithms library. http://www.cgal.org.

[3] 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.

[4] I. Z. Emiris, E. P. Tsigaridas, and G. M. Tzoumas. Exact Delaunay graph of smooth convex pseudo-circles: General predicates, and implementation for ellipses. Available from http://www.di.uoa.gr/∼geotz/, 2008.

[5] I. Z. Emiris, E. P. Tsigaridas, and G. M. Tzoumas. Predicates for the Exact Voronoi Diagram of Ellipses under the Euclidean Metric. *Int. J. Comp. Geom. & Apps, Special Issue SoCG'06*, 18(6):567–597, 2008.

[6] I. Z. Emiris and G. M. Tzoumas. Exact and efficient evaluation of the InCircle predicate for parametric ellipses and smooth convex objects. *Comput. Aided Des.*, 40(6):691–700, 2008.

[7] 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).

[8] M. Held and S. Huber. Topology-Oriented Incremental Computation of Voronoi Diagrams of Circular Arcs and Straight Line-Segments. *Computer Aided Design*, (to appear), 2008. Special issue on Voronoi diagrams.

[9] M. I. Karavelas and M. Yvinec. Voronoi diagram of convex objects in the plane. In *Proc. Europ. Symp. Algorithms*, LNCS, pages 337–348. Springer, 2003.

[10] R. Ramamurthy and R. Farouki. Voronoi diagram and medial axis algorithm for planar domains with curved boundaries - II: detailed algorithm description. *J. Comput. Appl. Math.*, 102(2):253–277, 1999.