Qualitative Symbolic Perturbation: a new geometry-based perturbation framework

Olivier Devillers, Menelaos Karavelas, Monique Teillaud

Qualitative Symbolic Perturbation: a new geometry-based perturbation framework

Olivier Devillers, Menelaos I. Karavelas, Monique Teillaud

ISSN 0249-6399   ISRN INRIA/RR--8153--FR+ENG

RESEARCH REPORT
N° 8153
December 2012
Project-Team Geometrica

Qualitative Symbolic Perturbation: a new geometry-based perturbation framework

Olivier Devillers*, Menelaos I. Karavelas†, Monique Teillaud‡

Project-Team Geometrica

Research Report n° 8153 — December 2012 — 28 pages

The work in this paper has been partially supported by the FP7-REGPOT-2009-1 project "Archimedes Center for Modeling, Analysis and Computation". M. Karavelas acknowledges support by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framerwork (NSRF) – Research Funding Program: THALIS – UOA (MIS 375891).

* Projet Geometrica, INRIA Sophia Antipolis - Méditerranée, France
† Department of Applied Mathematics, University of Crete, Greece
‡ Projet Geometrica, INRIA Sophia Antipolis - Méditerranée, France

RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Abstract: In the literature, the generic way to address degeneracies in computational geometry is the *Symbolic Perturbation* paradigm: the input is made dependent of some parameter ε so that for ε positive and close to zero, the input is close to the original input, while at the same time, in non-degenerate position. A geometric predicate can usually be seen as the sign of some function of the input. In the symbolic perturbation paradigm, if the function evaluates to zero, the input is perturbed by a small positive ε, and the sign of the function evaluated at the perturbed input is used instead.

The usual way of using this approach is what we will call *Algebraic Symbolic Perturbation* framework. When the function to be evaluated is a polynomial of the input, its perturbed version is seen as a polynomial in ε, whose coefficients are polynomials in the input. These coefficients are evaluated by increasing degree in ε until a non-vanishing coefficient is found. The number of these coefficients can be quite large and expressing them in an easily and efficiently computable manner (e.g., factorized) may require quite some work.

We propose to address the handling of geometric degeneracies in a different way, namely by means of what we call the *Qualitative Symbolic Perturbation* framework. We no longer use a single perturbation that must remove all degeneracies, but rather a sequence of perturbations, such that the next perturbation is being used only if the previous ones have not removed the degeneracies. The new perturbation is considered as *symbolically smaller* than the previous ones. This approach allows us to use simple elementary perturbations whose effect can be analyzed and evaluated: (1) by geometric reasoning instead of algebraic development of the predicate polynomial in ε, and (2) independently of a specific algebraic formulation of the predicate.

We apply our framework to predicates used in the computation of Apollonius diagrams in 2D and 3D, as well as the computation of trapezoidal maps of circular arcs.

Key-words: Degeneracies, Robustness issues, Apollonius diagram, Trapezoidal map.

Perturbation symbolique qualitative: une nouvelle méthode de perturbation basée géométrie

Résumé : L'approche généralement utilisée pour s'abstraire de l'existence de configurations dégénérées en géométrie algorithmique est la méthode des *perturbations symboliques :* les données sont rendues dépendantes d'un paramètre ε tel que pour ε positif et proche de zéro les données soient proches des données originales mais en garantissant d'être en position générale. Un prédicat géométrique peut souvent être vu comme le signe d'une fonction des données. Dans la méthode de perturbation symbolique, si l'évaluation de la fonction donne zéro, les données sont perturbées avec un petit ε positif et le signe de la fonction évaluée sur les données perturbées est utilisé comme réponse au prédicat.

L'application habituelle de cette approche est ce que nous appellerons *perturbations symboliques algébriques.* Quand la fonction a évaluer est un polynôme, sa version perturbée peut être vue comme un polynôme en ε dont les coefficients sont des polynômes dans les données. Ces coefficients sont alors évaluées par degré croissant en ε, jusqu'à ce qu'un coefficient qui ne soit pas évalué à zéro soit trouvé. Le nombre de ces coefficients peut être assez grand et les exprimer sous une forme permettant une évaluation facile et rapide (par exemple factorisée) peut représenter un travail important.

Nous proposons de résoudre les dégénérescences géométriques d'une manière différente que nous appellerons *perturbations symboliques qualitatives.* Au lieu d'une unique perturbation qui doit résoudre toutes les dégénérescences, nous utilisons une séquence de perturbations telle que la perturbation suivante n'est utilisée que si les précédentes n'ont pas supprimé les dégénérescences. La nouvelle perturbation est *symboliquement plus petite* que les précédentes. Cette approche permet d'utiliser des perturbations très simples dont l'effet peut être analysé et évalué (1) par un raisonnement géométrique plutôt que par un développement algébrique du polynôme en ε, et (2) de manière indépendante d'une formulation algébrique spécifique du prédicat.

Nous utilisons notre méthode pour le calcul des diagrammes d'Apollonius en 2D et 3D, ainsi que pour le calcul de la carte des trapèzes d'arcs de cercles.

Mots-clés : Dégénérescences, robustesse algorithmique, Diagramme d'Apollonius, carte des trapèzes

1 Introduction

In earlier times of computational geometry, the treatment of degenerate configurations was mainly ignored and papers often started with a sentence claiming "we assume that no four points are co-circular" or "no five spheres are tangent to a common line". This allowed for the design and analysis of data structures and algorithms without taking care of special cases.

Symbolic perturbations have been introduced to handle degenerate situations, which actually occur in practice [14, 13, 7]. When a geometric algorithm or data structure has originally been designed without addressing degeneracies, the use of a symbolic perturbation allows it to still operate on degenerate cases.

Let $G(\boldsymbol{x})$ be a geometric structure defined when the input data $\boldsymbol{x}$ satisfies some non-degeneracy assumptions, and let $\boldsymbol{x}_0$ be some input that is degenerate for G. A symbolic perturbation consists in modifying the input $\boldsymbol{x}$ as a continuous function $\pi(\boldsymbol{x}_0, \varepsilon)$ of a parameter ε. This is done in such a way that for $\varepsilon = 0$, the modified input $\pi(\boldsymbol{x}_0, 0)$ is equal to $\boldsymbol{x}_0$, and $\pi(\boldsymbol{x}_0, \varepsilon)$ is non-degenerate for G for sufficiently small positive values of ε. In that case the structure $G(\boldsymbol{x}_0)$ is defined as the limit of $G(\pi(\boldsymbol{x}_0, \varepsilon))$ when $\varepsilon \to 0^+$.

A symbolic perturbation allows an algorithm that computes $G(\boldsymbol{x})$ in generic situations to compute $G(\boldsymbol{x}_0)$ for the degenerate input $\boldsymbol{x}_0$. Most decisions taken by the algorithm are usually taken by looking at *geometric predicates*, which are signs of functions of the input. The original algorithm assumes that these functions never return 0. When applying a symbolic perturbation, a predicate $\text{sign}(p)$ evaluated at $\boldsymbol{x}_0$ returns the limit of $\text{sign}(p(\pi(\boldsymbol{x}_0, \varepsilon)))$ as $\varepsilon \to 0^+$. The sign of $p(\boldsymbol{x}_0)$ can thus be evaluated, provided that $p(\pi(\boldsymbol{x}_0, \varepsilon))$ is not identically equal to 0 in a (right) neighborhood of $\varepsilon = 0$.

In previous works [7, 8, 12, 1, 6], a predicate is the sign of a polynomial P in some input $\boldsymbol{x} \in \mathbb{R}^m$. The input $\boldsymbol{x}$ is perturbed as an element $\pi(\boldsymbol{x}_0, \varepsilon)$ of $\mathbb{R}^m$ whose coordinates are polynomials in $\boldsymbol{x}_0$ and ε, such that $\pi(\boldsymbol{x}_0, \varepsilon)$ goes to $\boldsymbol{x}_0$ when $\varepsilon \to 0^+$, and the perturbed predicate needs to evaluate somehow the limit $\lim_{\varepsilon \to 0^+} \text{sign}(P(\pi(\boldsymbol{x}_0, \varepsilon)))$. Since P is a polynomial, $P(\pi(\boldsymbol{x}_0, \varepsilon))$ can be rewritten as a polynomial in ε whose monomials in ε are ordered in terms of increasing degree. The first coefficient is actually $P(\boldsymbol{x}_0)$, and the signs of the following coefficients can be viewed as secondary predicates on $\boldsymbol{x}_0$. The coefficients of $P(\pi(\boldsymbol{x}_0, \varepsilon))$ are evaluated in increasing degrees in ε, until a non-vanishing coefficient is found. The sign of this coefficient is then returned as the value of the predicate $\text{sign}(P(\boldsymbol{x}_0))$. A perturbation scheme is said to be *effective* for a predicate $\text{sign}(P)$ if for any $\boldsymbol{x}_0$ the polynomial $P(\pi(\boldsymbol{x}_0, \varepsilon))$ is never the zero polynomial. The main difficulty when designing a perturbation scheme for $G(\boldsymbol{x})$ is to find a function $\pi(\boldsymbol{x}_0, \varepsilon)$, such that the perturbation scheme can be proved to be effective, and the perturbed predicate is easy to evaluate, e.g., using small degree polynomials.

Contribution. In this paper we propose a new framework for resolving degenerate configurations in geometric computing. Unlike classical algebraic symbolic perturbation techniques, we do not rely on a specific algebraic formulation of the predicate, but rather consider it in geometric terms. Instead of having a single perturbation parameter that governs the way the input objects and/or predicates are modified, we allow for a sequence of perturbation parameters and resolve degeneracies in a purely geometric manner, and independently of a specific algebraic formulation of the predicate. For this reason, our technique is particularly suitable for predicates whose algebraic description is not unique or too complicated. Conceptually, we symbolically perturb the input objects one-by-one, using a well-defined canonical ordering that corresponds to considering first the object that is perturbed most. The objects are symbolically perturbed until the degenerate configuration ceases to be degenerate, at which point the degeneracy of the predicate is resolved.

Standard algebraic symbolic perturbation schemes provide us automatically with the auxiliary predicates that we need to evaluate. These predicates are, by design, of at most the same algebraic degree as the original predicate, but evaluating them in an efficient manner (e.g., factorized) is far from being an obvious task. Qualitative symbolic perturbation schemes cannot guarantee that the auxiliary predicates are not more complicated algebraically (i.e., of higher algebraic degree) from the original predicate; however, in principle, the auxiliary predicates that we have to deal with are expected to be more tangible, since their analysis is based on geometric considerations.

The number of objects that needs to be perturbed depends on the specific predicate that we analyze (for example in the 2D Apollonius diagram perturbing a single object, for a given predicate, always suffices, whereas in its 3D counterpart, we may need to perturb two input objects). The goal in this framework is to devise an appropriate sequence of perturbations which guarantees that eventually, i.e., after having perturbed sufficiently many input objects, the degenerate predicate is resolved in a non-degenerate manner.

In the next section of the paper we formally define our qualitative symbolic perturbation framework. In Sections 3 and 4 we describe such qualitative perturbation schemes for the main predicates of two geometric structures: (1) the 2D/3D Apollonius diagram, and (2) the 2D arrangement of circular arcs. We end with Section 5, where we discuss the advantages and disadvantages of our framework, and indicate directions for future research.

2 General framework

Let $G(\boldsymbol{x})$ be a geometric structure whose computation depends on a predicate $\operatorname{sign}(p(\boldsymbol{x}))$, where p is a real function. We design the perturbation scheme π as a sequence of successive perturbations π_i, $0 \leq i < N$, with $\pi(\boldsymbol{x}, \boldsymbol{\varepsilon}) = \pi_0(\pi_1(\pi_2(\dots \pi_{N-1}(\boldsymbol{x}, \varepsilon_{N-1}) \dots, \varepsilon_2), \varepsilon_1), \varepsilon_0)$, where $\boldsymbol{\varepsilon} = (\varepsilon_0, \varepsilon_1, \varepsilon_2, \dots, \varepsilon_{N-1}) \in \mathbb{R}^N$. The number of perturbations N is part of the perturbation scheme and usually depends on the input size. The perturbations are examined by decreasing order of magnitude, and ε_i is considered much bigger than ε_j if $i > j$. Since $\boldsymbol{\varepsilon}$ is no longer a single real number, we have to determine how the limit is taken; we thus define $G(\boldsymbol{x})$ to be the limit:

$$G(\boldsymbol{x}) = \lim_{\varepsilon_{N-1} \to 0^+} \lim_{\varepsilon_{N-2} \to 0^+} \cdots \lim_{\varepsilon_1 \to 0^+} \lim_{\varepsilon_0 \to 0^+} G(\pi(\boldsymbol{x}, \boldsymbol{\varepsilon})).$$

The perturbed predicate

$$\lim_{\varepsilon_{N-1} \to 0^+} \lim_{\varepsilon_{N-2} \to 0^+} \cdots \lim_{\varepsilon_1 \to 0^+} \lim_{\varepsilon_0 \to 0^+} \operatorname{sign}(p(\pi(\boldsymbol{x}, \boldsymbol{\varepsilon})))$$

is evaluated by first computing $p(\pi(\boldsymbol{x}, (0, 0, \dots, 0)))) = p(\boldsymbol{x})$, and returning its sign if it is defined.

If $p(\boldsymbol{x}) = 0$, we look at the function $p(\pi(\boldsymbol{x}, (0, 0, \dots, \varepsilon_{N-1}))) = p(\pi_{N-1}(\boldsymbol{x}, \varepsilon_{N-1}))$; if this function is not vanishing when ε_{N-1} lies in a sufficiently small neighborhood to the right of 0, its sign can be returned. More formally, we compute the limit

$$\lim_{\varepsilon_{N-1} \to 0^+} \operatorname{sign}(p(\pi_{N-1}(\boldsymbol{x}, \varepsilon_{N-1}))).$$

If this limit is non-zero, it is returned as the value of the predicate $\operatorname{sign}(p(\boldsymbol{x}))$; otherwise, when $\varepsilon_{N-1} \geq 0$ lies in a neighborhood to the right of 0, we have $p(\pi_{N-1}(\boldsymbol{x}, \varepsilon_{N-1})) = 0$, in which case we have to further perturb our geometric input. We examine the limit

$$\lim_{\varepsilon_{N-1} \to 0^+} \lim_{\varepsilon_{N-2} \to 0^+} \operatorname{sign}(p(\pi_{N-2}(\pi_{N-1}(\boldsymbol{x}, \varepsilon_{N-1}), \varepsilon_{N-2}))). \tag{1}$$

but since the perturbed input is degenerate for all values of ε_{N-1} in a neighborhood to the right of 0, it is usually equivalent to look at the limit

$$\lim_{\varepsilon_{N-2}\to 0^+} \operatorname{sign}(p(\pi_{N-2}(\boldsymbol{x}, \varepsilon_{N-2}))). \tag{2}$$

More formally, the requirement that allows to reduce the evaluation of the limit in (1) to that in (2) is the following: for all $\mu < N$, if $p(\pi(\boldsymbol{x}, (0, 0, \ldots, 0, \varepsilon_{\mu+1}, \varepsilon_{\mu+2}, \ldots, \varepsilon_{N-1})))$ is the zero polynomial, then the sign of $p(\pi(\boldsymbol{x}, (0, 0, \ldots, 0, \varepsilon_{\mu}, \varepsilon_{\mu+1}, \varepsilon_{\mu+2}, \ldots, \varepsilon_{N-1})))$ does not depend on $\varepsilon_{\mu+1}, \varepsilon_{\mu+2}, \ldots, \varepsilon_{N-1}$.

Under such a hypothesis, a perturbation scheme following the qualitative symbolic perturbation framework consists in successively computing the limits

$$\lim_{\varepsilon_{\nu}\to 0^+} \operatorname{sign}(p(\pi_{\nu}(\boldsymbol{x}, \varepsilon_{\nu}))),$$

for ν going down from $N-1$ to 0, until a non-zero value is found. To assert that the perturbation is effective, we need to prove that we actually can find such a non-zero limit.

In the case where the predicate is a polynomial, as in algebraic symbolic perturbations we get a sequence of successive evaluations; however, the expressions that need to be evaluated have been obtained in a different way and are *a priori* different. The main advantage of this approach is that we may use a very simple perturbation π_ν, since we do not need each perturbation π_ν to be effective, but rather the composed perturbation π. The simplicity of π_ν allows us to look at the limit in a geometric manner instead of algebraically computing some appropriate coefficient of $p(\pi(\boldsymbol{x}, \varepsilon))$. In some previous works, the algebraic symbolic perturbation framework was proved to be effective by a careful choice of the exponents for ε, so as to make some terms negligible. The same solution is used here in a more geometrically explicit way. If one insists on viewing the qualitative symbolic perturbation the traditional way, we can make all the ε_ν's dependent on a single parameter κ that plays the traditional role of ε. It is enough to take ε_ν exponentially increasing with respect to ν. For example one such choice can be to set $\varepsilon_{N-1} = \kappa$, and $\varepsilon_\nu = \left(\exp\left(\frac{1}{\varepsilon_{\nu+1}}\right)\right)^{-1}$, for $0 \leq \nu < N-1$.

3 The Apollonius diagram

3.1 Definition

The *Apollonius diagram*, also known as *additively weighted Voronoi diagram*, is defined on a set of weighted points in the Euclidean space $\mathbb{R}^d$. The Euclidean distance is denoted as $|\cdot|$. The *weighted distance* from a query point q to a weighted point (p, w), where p is a point in the Euclidean space and $w \in \mathbb{R}$, is $|pq| - w$. The Apollonius diagram is the closest point diagram for this distance. It generalizes the Voronoi diagram, defined on non-weighted points.

Given a set of weighted points, also called *sites*, it is clear that adding the same constant to all weights does not change the Apollonius diagram. Thus, in the sequel, we may freely translate the weights to ensure, for example, that all weights are positive, or that a particular weight is zero. A site $(s, w), w \geq 0$ can be identified to the sphere S centered at s and of radius w. The distance from a query point q to a site (s, w) is the Euclidean distance from q to S, with a negative sign if q lies inside S. Figure 1 gives an example of a two-dimensional Apollonius diagram.

An Apollonius vertex v is a point at the same distance from $d+1$ sites $S_0, S_1, \ldots, S_d$ in general position. We call the configuration *external* if v is outside the S_i, and *internal* if it is inside. If the configuration is external (resp., internal), v is the center of a sphere externally (resp., internally) tangent to the sites S_i (see green (resp., dark green) disks in the figure). It is

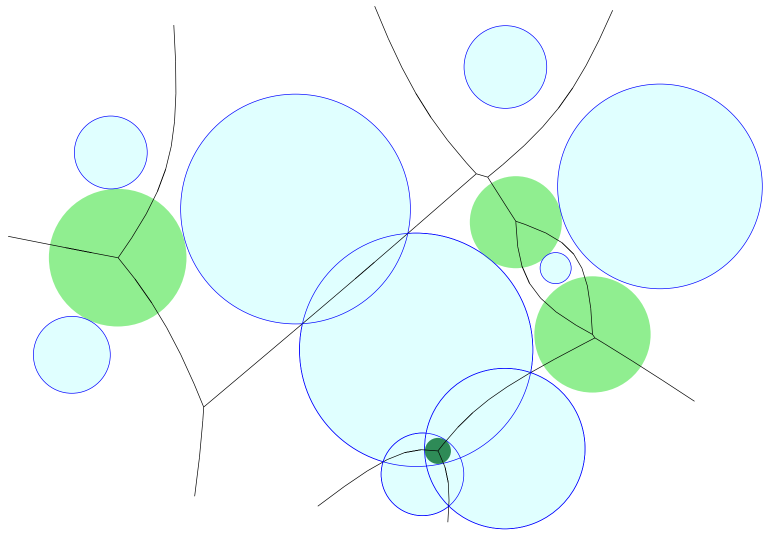

Figure 1: Planar Apollonius diagram. Weighted points are in light blue, the green disks are centered at an Apollonius vertex and their radius is the weighted distance to its closest neighbors. The darker green disk actually has a negative distance to its closest neighbors.

always possible to ensure an external configuration by adding a suitable constant to the weights of all S_i, such that all weights are non-negative, while the smallest among them is equal to zero.

An Apollonius vertex is actually defined by a sequence of $d+1$ sites in general position, up to a positive permutation of the sequence. To prove this fact, we first observe that $d+1$ sites in general position define zero, one or two Apollonius vertices. First assume that all weights are non-negative and $w_0 = 0$, so as to be in external configuration. Consider now an inversion transformation (see Figure 2) with the point s_0 as the pole. The point s_0 goes to infinity, while the spheres S_i, $i = 1, \ldots, d$ become new spheres $Z_i = (z_i, \rho_i)$. Determining the spheres B_α tangent to the spheres S_i, $i = 0, \ldots, d$, is equivalent to determining the hyperplanes Π_α tangent to the spheres Z_i, $i = 1, \ldots, d$, with all spheres on the same side of Π_α (where α indexes the different solutions). The fact that B_α is externally tangent to the S_i is equivalent to the fact that Π_α separates the Z_i from the origin. The normalized equation of Π_α: $\lambda_\alpha \cdot x + \delta_\alpha = 0$, with $|\lambda_\alpha| = 1$, gives the signed distance of a point x from Π_α. We have

$$\Pi_\alpha \text{ tangent to } Z_i,\ 1 \le i \le d \quad \Longleftrightarrow \quad \begin{cases} \lambda_\alpha \cdot z_i + \delta_\alpha = \rho_i,\ 1 \le i \le d \\ |\lambda_\alpha| = 1 \end{cases} .$$

This system of one quadratic and d linear equations has at most two real solutions by Bézout's theorem. Depending on the position of the origin with respect to Π_α (or equivalently of the sign of δ_α), zero, one or the two solutions may correspond to external configurations. If there are two solutions Π_α and Π_β, observe that they are symmetric with respect to the hyperplane spanned by the z_i, thus the d-simplex formed by the tangency points and the origin has different orientation for the two solutions. This implies that the two solutions can be distinguished by the signature of the permutation of the S_i.

Finally, in the inverted space, the general position hypothesis means that the Z_i do not have an infinity of tangent hyperplanes; the latter can occur only if the z_i (and thus the s_i) are affinely dependent.

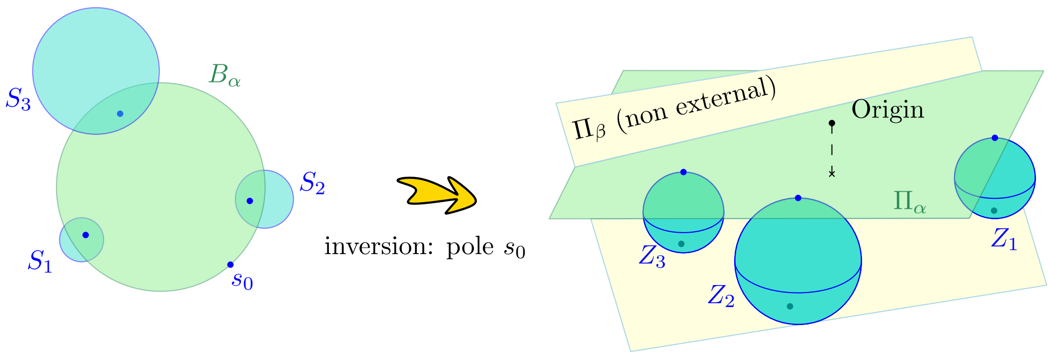

Figure 2: The spheres B_α externally tangent to the sites S_i, $i = 0, \ldots, d$, correspond, via the inversion transformation with s_0 as the pole, to hyperplanes Π_α tangent to the spheres Z_i that separate the Z_i from the origin.

3.2 The VertexConflict predicate

The predicate $\mathsf{VertexConflict}(\boldsymbol{S}^v, Q)$ answers the following question:

> *Does an Apollonius vertex v defined, up to a positive permutation, by a $(d+1)$-tuple of sites $\boldsymbol{S}^v = (S_{i_0}, S_{i_1}, \ldots, S_{i_d})$ remain as a vertex of the diagram after another site Q is added?*

If the site centered at v and tangent to the sites of the tuple $\boldsymbol{S}^v$ is in internal configuration, we can add a negative constant to the radii of all spheres in $\boldsymbol{S}^v \cup \{Q\}$ so that the smallest site in $\boldsymbol{S}^v$ has zero radius. Then the configuration of the common tangent sphere becomes external. In this manner, we can always restrict our analysis to the case where the Apollonius vertex we consider is in external configuration. Note that this may lead to a negative weight for Q, which is treated below.

We denote by $B_{i_0 i_1 \ldots i_d}$ the open ball whose closure $\overline{B}_{i_0 i_1 \ldots i_d}$ is tangent to the sites of $\boldsymbol{S}^v$. The contact points $t_{i_0}, t_{i_1}, \ldots, t_{i_d}$ define a positively oriented d-simplex.

If Q has a non-negative weight, the predicate $\mathsf{VertexConflict}(\boldsymbol{S}^v, Q)$ answers (see Figure 3)

- "conflict" if Q intersects $B_{i_0 i_1 \ldots i_d}$,
- "no conflict" if Q and $\overline{B}_{i_0 i_1 \ldots i_d}$ are disjoint,
- "degenerate" if Q and $B_{i_0 i_1 \ldots i_d}$ do not intersect, while Q and $\overline{B}_{i_0 i_1 \ldots i_d}$ are tangent.

If $Q = (p, -w), w \geq 0$, has a non-positive weight, we define Q^- as the sphere centered at p with radius w. Then $\mathsf{VertexConflict}(\boldsymbol{S}^v, Q)$ answers

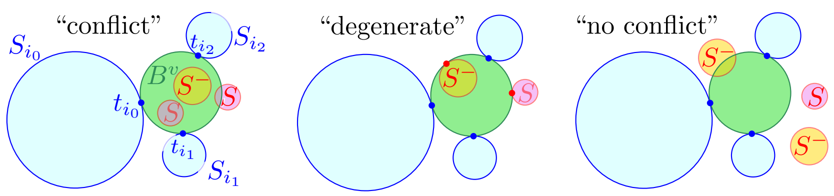

Figure 3: Examples of positions of Q or Q^- that are in "conflict", "degenerate" or "no conflict" configuration with an Apollonius vertex v.

- "conflict" if Q^- is included in $B_{i_0 i_1 \ldots i_d}$,
- "no conflict" if Q^- intersects the complement of $\overline{B}_{i_0 i_1 \ldots i_d}$,
- "degenerate" if Q^- is included in $\overline{B}_{i_0 i_1 \ldots i_d}$ and is tangent to its boundary.

3.3 Perturbing circles for the 2D Apollonius diagram

3.3.1 Algebraic perturbation for the VertexConflict predicate

Before using our qualitative symbolic perturbation framework to design the perturbed predicate, we briefly sketch how a standard algebraic perturbation framework could be applied.

Let S_i, S_j, S_k be the three sites that define an Apollonius circle in the Apollonius diagram and let $Q = S_q$ be the query site. In the algebraic formulation of the predicate by Emiris and Karavelas [9], the evaluation of $\mathsf{VertexConflict}(S_i, S_j, S_k, Q)$ relies on the computation of the sign of the quantity:

$$I := E_{xw} E_x + E_{yw} E_y + E_{xy} \sqrt{\Delta}, \qquad \Delta = (E_x)^2 + (E_y)^2 - (E_w)^2,$$

where

$$E_s = \begin{vmatrix} s_j^* & p_j^* \\ s_k^* & p_k^* \end{vmatrix}, \qquad E_{st} = \begin{vmatrix} s_j^* & t_j^* & p_j^* \\ s_k^* & t_k^* & p_k^* \\ s_q^* & t_q^* & p_q^* \end{vmatrix}, \qquad s, t \in \{x, y, w\},$$

and

$$x_\nu^* = x_\nu - x_i, \quad y_\nu^* = y_\nu - y_i, \quad w_\nu^* = w_\nu - w_i, \quad p_\nu^* = (x_\nu^*)^2 + (y_\nu^*)^2 - (w_\nu^*)^2, \quad \nu \in \{j, k, q\}.$$

The quantity I is a quantity of the form $X_0 + X_1\sqrt{Y}$, where the algebraic degrees of X_0, X_1 and Y are 7, 4 and 6, respectively. Its sign can be computed by means of the formula

$$\operatorname{sign}(X_0 + X_1\sqrt{Y}) = \begin{cases} \operatorname{sign}(X_0) & \text{if } Y = 0 \\ \operatorname{sign}(X_1) & \text{if } X_0 = 0 \\ \operatorname{sign}(X_0) & \text{if } X_1 = 0 \\ \operatorname{sign}(X_0) & \text{if } \operatorname{sign}(X_0) = \operatorname{sign}(X_1) \\ \operatorname{sign}(X_0)\operatorname{sign}(X_0^2 - X_1^2 Y) & \text{otherwise} \end{cases}. \tag{3}$$

It is thus concluded in [9, Theorem 11] that the algebraic degree of the predicate is 14.

If fact we can further decrease the algebraic degree of the predicate by observing that the quantity $X_0^2 - X_1^2 Y$ can be factorized as follows (see Section A.1 of the Appendix for the details of this derivation):

$$X_0^2 - X_1^2 Y = [(E_x)^2 + (E_y)^2]\,[(E_{xw})^2 + (E_{yw})^2 - (E_{xy})^2],$$

where the first factor is a non-negative quantity of degree 6, whereas the second factor is of degree 8. In fact, when we compute the sign of the quantity $X_0^2 - X_1^2 Y$, we already know that the quantity $(E_x)^2 + (E_y)^2$ is strictly positive, since otherwise X_0 would have been zero (X_0 is a linear combination of E_x and E_y), which has already been ruled out according to the procedure in rel. (3). Hence the algebraic degree of the predicate is 8.

If $S_\nu = (x_\nu, y_\nu, w_\nu)$ is perturbed in $S_\nu^\varepsilon = (x_\nu, y_\nu, w_\nu + \varepsilon_\nu)$ for $\nu \in \{i, j, k, q\}$, then developing an expression like $(E_{xw})^2$ will give a polynomial of degree 6 in ε_i, ε_j, ε_k and ε_q with 186 terms. Assigning ε_i, ε_j, ε_k and ε_q to be a polynomial function of a single variable ε (for example, we may set $\varepsilon_\nu = \varepsilon^{\alpha_\nu}$, $\nu \in \{i, j, k, q\}$) transforms the expression in a univariate polynomial in ε.

When performing such an assignment, either some of the terms collapse making their geometric and algebraic interpretation difficult, or α_i, α_j, α_k and α_q have to be chosen carefully so that the coefficients of the various monomials of the ε_ν's in the resulting polynomial do not collapse. Even if can find an assignment that does not make the coefficients (of the originally different terms) to collapse, we are still faced with the problem of analyzing the monomials, and, by employing algebraic and/or geometric arguments, showing that there is at least one coefficient of the polynomial that does not vanish.

3.3.2 Qualitative perturbation of the VertexConflict predicate

Our perturbation depends on some given ordering of the sites. Each site $S_i = (x_i, y_i, w_i)$ is perturbed in $S_i^\varepsilon = (x_i, y_i, w_i + \varepsilon_i)$, $\varepsilon_i \geq 0$ with S_j perturbed before S_k if $j > k$. In other words, if the configuration is degenerate, the site of largest index grows much more than the others, which can be considered as fixed. Following our qualitative symbolic perturbation framework, if after we have enlarged the site of maximum index, the configuration is still degenerate, we enlarge the site with the second largest index, and so on. In fact, we need only consider the sites involved in the predicate, and enlarge them one-by-one, in decreasing order, until the resulting configuration is non-degenerate, in which case the predicate can be resolved.

Among all possible indexing choices for the sites, we are going to choose what we call the *max-weight* ordering that assigns a larger index to the site with larger weight. As a result, a site with larger weight is perturbed more, and in order to resolve the predicate we need to consider the sites in order of decreasing weight, until the degeneracy is resolved. To break ties between sites with the same weights, we use the lexicographic ordering of their centers: among two sites with the same weight, the site whose center is smaller lexicographically than the other is assigned a smaller max-weight order.

The max-weight ordering has a strong advantage compared to any other possible ordering of the sites: it favors sites with larger weight, so, if two sites are internally tangent the site with the larger weight will be perturbed first, in which case the site with the smallest weight will be inside the interior of the other site, and its Apollonius region will disappear. As a first consequence, this ordering minimizes the number of Apollonius regions in the diagram, or, equivalently it maximizes the number of hidden sites in the diagram. Secondly, and most importantly, the tangency points of the sites with the Apollonius circles that they define in the diagram are pairwise distinct. This property makes the analysis of the perturbed predicates much simpler, whereas the Apollonius diagram computed does not exhibit pathological cases, such as Apollonius regions with empty interiors and boundaries that degenerate to line segments and half-lines.

Let us denote by q the max-weight order of Q. If $q > i, j, k$ and $\mathsf{VertexConflict}(S_i, S_j, S_k, Q) =$ "degenerate", it is clear that the perturbed predicate is equal to

$$\mathsf{VertexConflict}^\varepsilon(S_i, S_j, S_k, Q) = \lim_{\varepsilon_q \to 0^+} \mathsf{VertexConflict}(S_i, S_j, S_k, Q^\varepsilon) = \text{“conflict”},$$

since Q is growing, while B_{ijk} can be considered as fixed(see Figure 4).

If q is not the largest index and $\mathsf{VertexConflict}(S_i, S_j, S_k, Q) =$ "degenerate", then B_{ijk} can be viewed as defined by three other circles among S_i, S_j, S_k, and Q. Since $B_{ijk} = B_{jki} = B_{kij}$, we can assume, without loss of generality, that $i > j, k, q$. Moreover, B_{ijk} coincides with either B_{jkq} or B_{kjq}, depending on the orientation of the tangency points of S_j, S_k and Q with B_{ijk}.

Since S_i is in conflict with B_{jkq} (or B_{kjq}) we simply need to determine if Q is also in conflict with respect to B_{ijk}. Let t_ν (resp., t_q) be the tangency point of S_ν (resp., Q) with B_{ijk}, $\nu \in \{i, j, k\}$, and notice that $t_i t_j t_k$ is a ccw triangle. We consider three cases depending on the position of t_q on ∂B_{ijk}. If t_q is different from t_i, t_j, and t_k, the four points form a convex quadrilateral. When perturbing S_i to become S_i^ε, the Apollonius vertex is split in two, which, in the dual, corresponds to a triangulation of the quadrilateral with vertices S_i, S_j, S_k, S_q. Since S_i is the most perturbed circle, the quadrilateral will be triangulated by linking S_i to the other three vertices. If t_q is on the same side as t_i with respect to the line $t_j t_k$, then the triangulation contains $S_i S_j S_k$ and, therefore, Q is not in conflict with B_{ijk}(see Figure 5), otherwise $S_i S_j S_k$ is not in the triangulation and Q has to be in conflict with B_{ijk}(see Figure 6).

If t_q is equal to t_i then, since $i > q$, Q is internally tangent to S_i and there is no conflict (S_i^ε contains Q in its interior, and thus Q has empty Apollonius region in the diagram). If t_q is equal to t_ν with $\nu \in \{j, k\}$ then either Q is internally tangent to S_ν, or S_ν is internally tangent to Q. In the former case, Q does not intersect the perturbed Apollonius disk B_{ijk}^ε and thus the result of the perturbed predicate is "no conflict"; in the latter case, Q intersects B_{ijk}^ε, and the perturbed predicate returns "conflict". Hence, in the case $t_q = t_\nu$, $\nu \in \{j, k\}$, the perturbed predicate returns "conflict" if an only if $q > \nu$.

3.3.3 Practical evaluation of the $\mathsf{VertexConflict}^\varepsilon$ predicate

Following the analysis in the previous section, $\mathsf{VertexConflict}^\varepsilon(S_i, S_j, S_k, Q)$ can be evaluated by the following procedure:

1. if $\mathsf{VertexConflict}(S_i, S_j, S_k, Q) \neq$ "degenerate", then return $\mathsf{VertexConflict}(S_i, S_j, S_k, Q)$;
2. if $q > i, j, k$, then return "conflict";
3. ensure that $i > j, k$ by a cyclic permutation of (i, j, k);
4. if $t_q = t_i$, then return "no conflict";
5. if $t_q = t_j$, then { if $q > j$ return "conflict", else return "no conflict" };
6. if $t_q = t_k$, then { if $q > k$ return "conflict", else return "no conflict" };
7. if $t_j t_k t_q$ is positively oriented, then return "no conflict", otherwise return "conflict".

Below, we describe in detail how the perturbed predicate is evaluated, and focus on two different ways for computing the orientation $\mathsf{Orientation}(t_j, t_k, t_q)$ of the tangency points, each having its advantages and disadvantages. Our aim is to reduce the complexity of the expressions to be evaluated, which is why we avoid evaluting the tangency points explicitly.

Step 1 is evaluated as described in Section 3.3.1. Steps 2 and 3 amount to sorting the indices of the four sites and determining if q is the largest, or, if this is not the case, finding the largest index. At Step 4, we already know that $i > q$, which implies that $w_i \geq w_q$, and hence the only possibility is that Q is internally tangent to S_i. So, in order to perform Step 4, we simply look at $p_q^* = (x_q - x_i)^2 + (y_q - y_i)^2 - (w_q - w_i)^2$: if $p_q^* = 0$, return "no conflict", otherwise continue with Step 5. Steps 5 and 6 can be resolved in analogous way: if $(x_q - x_\nu)^2 + (y_q - y_\nu)^2 - (w_q - w_\nu)^2 = 0$, then if $q > \nu$ (resp. $q < \nu$), we return "conflict" (resp. "no conflict"). Otherwise, we continue with the last step of the procedure.

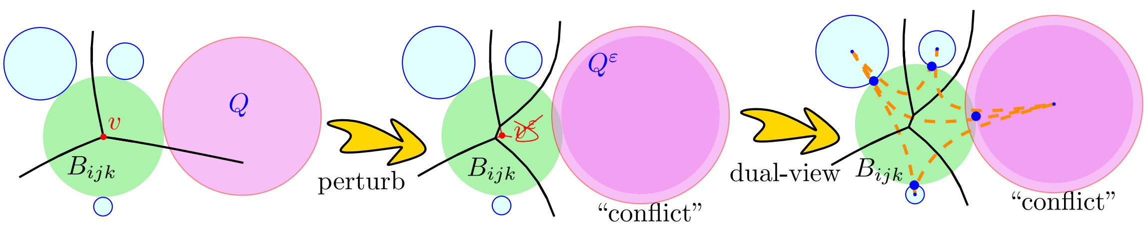

Figure 4: Perturbing a degenerate Apollonius vertex: the case $q > i, j, k$.

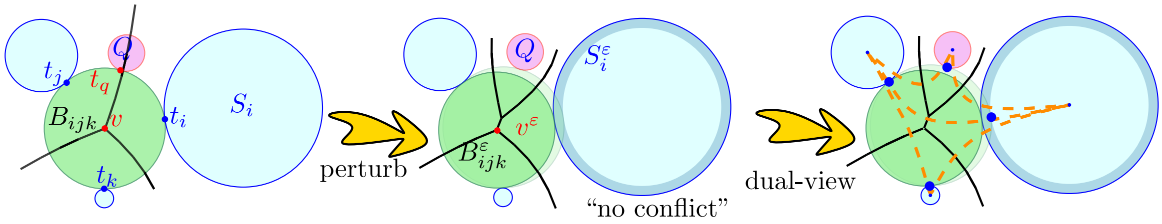

Figure 5: Perturbing a degenerate Apollonius vertex: the case $i > j, k, q$ and $t_j t_k t_q$ is a ccw triangle.

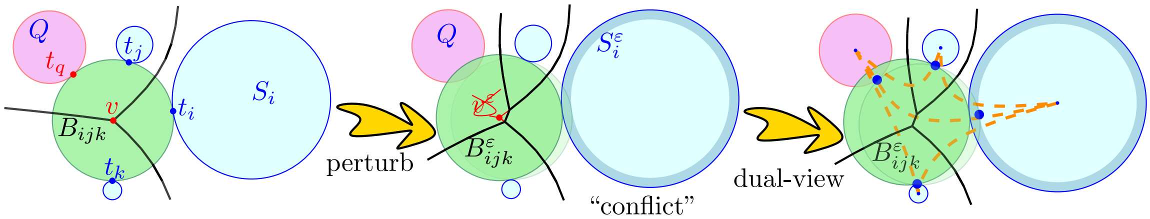

Figure 6: Perturbing a degenerate Apollonius vertex: the case $i > j, k, q$ and $t_j t_k t_q$ is a cw triangle.

Step 7 requires the evaluation of $\mathsf{Orientation}(t_j, t_k, t_q)$. We can do this in any of the following two ways.

(a) First, we essentially follow the same procedure that was used in [9] for evaluating the $\mathsf{VertexConflict}$ predicate. The various calculations described below may be found in Section A.2 of the Appendix. We decrease the weights of S_i, S_j, S_k and S_q, which results in S_i becoming a point, while the weights of S_j, S_k and S_q become non-positive. We define $S_j^- = (s_j, -w_j), S_k^- = (s_k, -w_k), S_q^- = Q^- = (s_q, -w_q)$. Using S_i as the pole, we invert S_j^-, S_k^- and Q^-, and let $Z_j^- = (u_j, v_j, \rho_j)$, $Z_k^- = (u_k, v_k, \rho_k)$ and $Z_q^- = (u_q, v_q, \rho_q)$ be the corresponding inverted sites (see Figure 7). Let L denote the line that is tangent to Z_j^-, Z_k^- and Z_q^-, and call ω_j, ω_k and ω_q the points of tangency of Z_j^-, Z_k^- and Z_q^- with L. Determining $\mathsf{Orientation}(t_j, t_k, t_q)$ is equivalent to determining if ω_q lies inside or outside the segment $\omega_j\omega_k$ (the cases $\omega_q = \omega_j$ and $\omega_q = \omega_k$ have been ruled out by Steps 5 and 6 of the predicate evaluation procedure): $\mathsf{Orientation}(t_j, t_k, t_q)$ is ccw if and only if t_q lies on B_{ijk} and on the arc delimited by t_j and t_k that contains t_i, which, in turn, is the case if and only if $\omega_q \in L$ lies outside the segment $\omega_j\omega_k$. To determine this, we consider the oriented lines $L_j^\perp$ and $L_k^\perp$ that are perpendicular to L and pass through the centers of Z_j^- and Z_k^- respectively. We assume that the positive orientation of $L_j^\perp$ and $L_k^\perp$ is in the direction of the (open) half-plane delimited by L that does not contain Z_j^- and Z_k^-. Let o_j and o_k be the result of the orientation test of (u_q, v_q) with respect to the two lines. If $o_j = o_k$, then ω_q lies outside the segment $\omega_j\omega_k$, and the predicate returns "no conflict" (see $Z_{Q_n}^-$ and $Z_{Q'_n}^-$ in Figure 7); otherwise, $o_j \neq o_k$, in which case ω_q lies inside the segment $\omega_j\omega_k$, and the predicate returns "conflict" (see $Z_{Q_c}^-$ in Figure 7).

Computing the orientation o_ν, $\nu \in \{j, k\}$ amounts to computing the sign of the quantity:

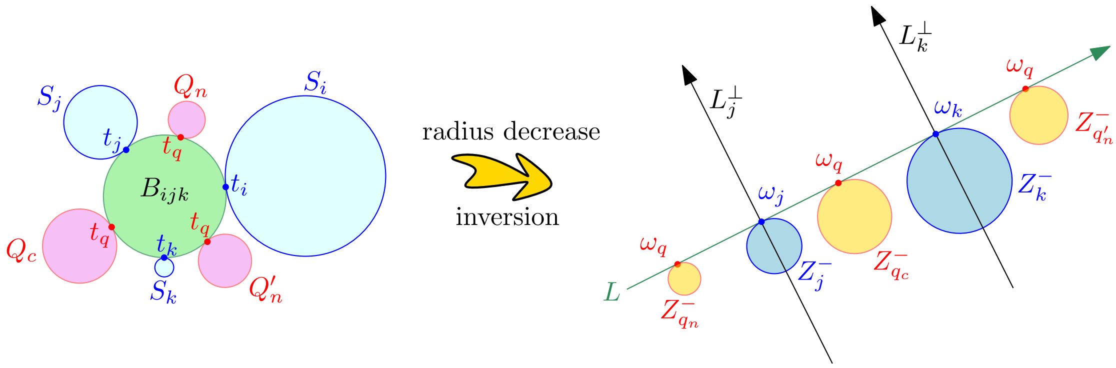

Figure 7: Computing the auxiliary predicate $\mathsf{Orientation}(t_j, t_k, t_q)$ using inversion. ω_{Q_c} lies inside $\omega_j\omega_k$, which corresponds to $\mathsf{Orientation}(t_j, t_k, t_{Q_c})$ = "cw". ω_{Q_n} (resp., $\omega_{Q'_n}$) lies outside $\omega_j\omega_k$, which corresponds to $\mathsf{Orientation}(t_j, t_k, t_{Q_n})$ = "ccw" (resp., $\mathsf{Orientation}(t_j, t_k, t_{Q'_n})$ = "ccw").

$$o'_\nu = p^*_\nu E_{xy} E_w + (E_x F_x + E_y F_y)\sqrt{\Delta}, \qquad F_s = \begin{vmatrix} s^*_q & p^*_q \\ s^*_\nu & p^*_\nu \end{vmatrix}, \qquad s \in \{x, y\}.$$

The sign can be resolved using the procedure in 3. At a first glance the algebraic degree of the predicate is 18. However, as for the $\mathsf{VertexConflict}$ predicate, we can factor the quantity $X_0^2 - X_1^2 Y$ as follows:

$$X_0^2 - X_1^2 Y = [(E_x)^2 + (E_y)^2]\, o''_\nu, \qquad o''_\nu = [(F_x)^2 + (F_y)^2]\,(E_w)^2 - (E_x F_x + E_y F_y)^2.$$

Since $(E_x)^2 + (E_y)^2$ cannot be zero (otherwise we would have resolved the sign of $X_0 + X_1\sqrt{Y}$ without resorting to the computing the sign of $X_0^2 + X_1^2 Y$), determining the sign of o'_ν reduces to determing the sign of $[(F_x)^2 + (F_y)^2]\,(E_w)^2 - (E_x F_x + E_y F_y)^2$, which is of algebraic degree 12. Notice that the way of evaluating $\mathsf{Orientation}(t_j, t_k, t_q)$ results in a perturbed predicate with higher algebraic degree than the unperturbed one. On the other hand, both o_j and o_k can be evaluated with extremely few operations in addition to those required for the $\mathsf{VertexConflict}$ predicate: observing that the quantities p^*_ν, E_x, E_y, E_w, $(E_w)^2$ and E_{xy} have already been computed when evaluating the $\mathsf{VertexConflict}$ predicate, and that F_x and F_y are minors of E_{xy}, and thus can be stored while evaluating E_{xy}, we need a maximum of 9 operations in order to compute the sign of o'_ν (3 ops for $E_x F_x + E_y F_y$, 3 ops for $(F_x)^2 + (F_y)^2$, 1 op for $(E_x F_x + E_y F_y)^2$, and another 2 ops for o''_ν).

(b) It has been shown in [9], that evaluating the orientation of three points where two are centers of sites and the third is an Apollonius vertex, can be performed using algebraic expressions of degree at most 14. In fact, this degree can again be decreased to 8 (see Section A.3 of the Appendix), in which case we can use this predicate to resolve $\mathsf{Orientation}(t_j, t_k, t_q)$ without resorting to a higher degree predicate.

As a first step we evaluate $o_1 = \mathsf{Orientation}(s_j, v_{ijk}, s_k)$, where v_{ijk} is the center of the Apollonius circles B_{ijk} of the three sites S_i, S_j, S_k. We perform this evaluation in order to determine whether the angle α of the ccw arc $\widehat{t_j t_k}$ on B_{ijk} is more or less that π. We distinguish between the following cases (see Figure 8):

o_1 = "collinear". In this case $\alpha = \pi$, and the line through t_j and t_k coincides with the line through s_j and s_k. Hence: $\mathsf{Orientation}(t_j, t_k, t_q) = \mathsf{Orientation}(s_j, s_k, s_q)$.

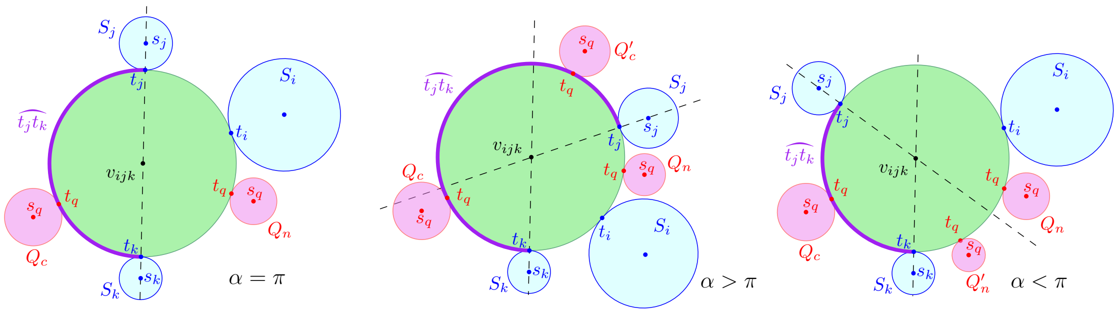

Figure 8: Computing the auxiliary predicate $\mathsf{Orientation}(t_j, t_k, t_q)$ using orientations involving the Apollonius vertex v_{ijk}. From left to right: the case $\alpha = \pi$, the case $\alpha > \pi$, and the case $\alpha < \pi$, where α is the angle of the ccw oriented arc $\widehat{t_j t_k}$.

o_1 = "ccw". In this case $\alpha > \pi$. We start by evaluating $o_2 = \mathsf{Orientation}(s_j, v_{ijk}, s_q)$. If $o_2 \neq$ "ccw" (see Q'_c in Figure 8(middle)), t_q lies to the right of the line through t_j and t_k, and thus $\mathsf{Orientation}(t_j, t_k, t_q)$ = "cw". Otherwise, we need to evaluate the orientation $o_3 = \mathsf{Orientation}(v_{ijk}, s_k, s_q)$; then $\mathsf{Orientation}(t_j, t_k, t_q)$ = "ccw" if and only if o_3 = "ccw" (see Q_n and Q''_c in Figure 8(middle)).

o_1 = "cw". In this case $\alpha < \pi$. We start by evaluating $o_2 = \mathsf{Orientation}(s_j, v_{ijk}, s_q)$. If $o_2 \neq$ "cw", t_q lies to the left of the line through t_j and t_k, and thus $\mathsf{Orientation}(t_j, t_k, t_q)$ = "ccw" (see Q_n in Figure 8(right)). Otherwise, we need to evaluate the orientation $o_3 = \mathsf{Orientation}(v_{ijk}, s_k, s_q)$; then $\mathsf{Orientation}(t_j, t_k, t_q)$ = "cw" if and only if o_3 = "cw" (see Q_c and Q'_n in Figure 8(right)).

The advantage of evaluating $\mathsf{Orientation}(t_j, t_k, t_q)$ as in (b), instead of as in (a), is that we rely on predicates of smaller algebraic degree (8 as opposed to 12). On the other hand, computing $\mathsf{Orientation}(t_j, t_k, t_q)$ as in (a), instead of as in (b), requires significantly less arithmetic operations to be performed.

3.3.4 Qualitative perturbation for the EdgeConflict predicate

The most important predicate in the computation of the 2D Apollonius diagram, and in fact any abstract Voronoi diagram, is the EdgeConflict predicate: given four sites S_i, S_j, S_k and S_l that define a Voronoi edge e in the diagram, and a query site Q, determine the type of conflict of Q with the edge e. This predicate is the basis of the randomized incremental construction algorithm for computing abstract Voronoi diagrams by Klein, Mehlhorn, and Meiser [11], as well as one of the main predicates analyzed by Emiris and Karavelas [9]. In [9] this predicate is decomposed to a number of subpredicates, one of them being the VertexConflict predicate.

We assume below that e lies on the bisector of S_i and S_j, oriented so that S_i is lying to the right of the bisector(refer also to Figure 9). The edge e inherits the orientation from its supporting bisector. The origin vertex of e is the Apollonius vertex defined by the (oriented) triple S_i, S_j and S_k, while the target vertex of e is the Apollonius vertex defined by the triple S_j, S_i and S_l. The EdgeConflict predicate determines the type of the subset of e that is destroyed by the insertion of Q in the Apollonius diagram of the four sites, and has six possible outcomes:

- "conflict origin": a subsegment of e adjacent to its origin vertex disappears in the Apollonius diagram of the five sites. This case occurs if and only if $\mathsf{VertexConflict}^{\varepsilon}(S_i, S_j, S_k, Q)$ =

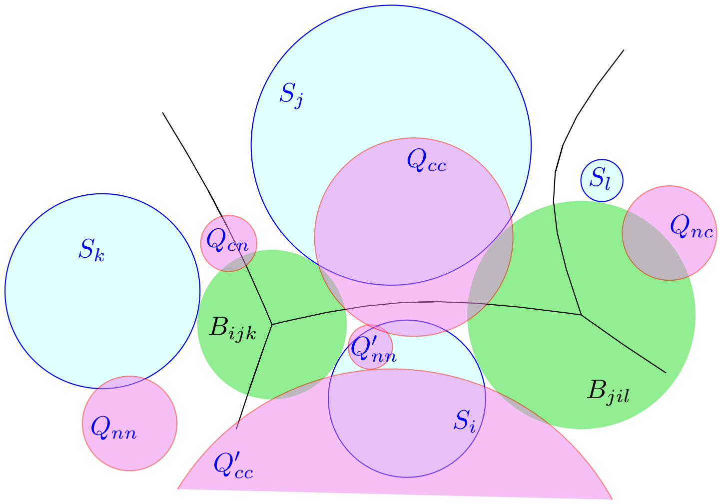

Figure 9: The different non-degenerate possible configurations for the $\mathsf{EdgeConflict}$ predicate.

"conflict" and $\mathsf{VertexConflict}^{\varepsilon}(S_j, S_i, S_l, Q)$ = "no conflict". This case is illustrated by Q_{cn} on Figure 9.

- "conflict target": is the symmetric case that occurs iff $\mathsf{VertexConflict}^{\varepsilon}(S_i, S_j, S_k, Q)$ = "no conflict" and $\mathsf{VertexConflict}^{\varepsilon}(S_j, S_i, S_l, Q)$ = "conflict". See Q_{nc} on Figure 9.
- "no conflict": no portion of e is destroyed by the insertion of Q in the Apollonius diagram of the four sites. This case can occur only when $\mathsf{VertexConflict}^{\varepsilon}(S_i, S_j, S_k, Q) = \mathsf{VertexConflict}^{\varepsilon}(S_j, S_i, S_l, Q)$ = "no conflict". See Q_{nn} on Figure 9.
- "conflict interior": a subsegment in the interior of e disappears in the Apollonius diagram of the five sites. This case can occur only when $\mathsf{VertexConflict}^{\varepsilon}(S_i, S_j, S_k, Q) = \mathsf{VertexConflict}^{\varepsilon}(S_j, S_i, S_l, Q)$ = "no conflict". See Q'_{nn} on Figure 9.
- "conflict entire edge": the entire edge e is destroyed by the addition of Q in the Apollonius diagram of the four sites. This case can occur only when $\mathsf{VertexConflict}^{\varepsilon}(S_i, S_j, S_k, Q) = \mathsf{VertexConflict}^{\varepsilon}(S_j, S_i, S_l, Q)$ = "conflict". See Q_{cc} on Figure 9.
- "conflict both": subsegments of e adjacent to its two vertices disappear in the Apollonius diagram of the five sites. This case can occur only when $\mathsf{VertexConflict}^{\varepsilon}(S_i, S_j, S_k, Q) = \mathsf{VertexConflict}^{\varepsilon}(S_j, S_i, S_l, Q)$ = "conflict". See Q'_{cc} on Figure 9.

Thus when the evaluations of predicate $\mathsf{VertexConflict}^{\varepsilon}$ on (S_i, S_j, S_k, Q) and (S_j, S_i, S_l, Q) are available, it only remains to distinguish between "no conflict" and "conflict interior", as well as between "conflict entire edge" and "conflict both". Assuming a non-degenerate configuration, this question is addressed in [9] using an auxiliary predicate of algebraic degree 16. The only situation where this auxiliary predicate has degeneracies is when $\mathsf{VertexConflict}(S_i, S_j, S_k, Q)$ or/and $\mathsf{VertexConflict}(S_j, S_i, S_l, Q)$ are "degenerate" and $\mathsf{VertexConflict}^{\varepsilon}(S_i, S_j, S_k, Q) = \mathsf{VertexConflict}^{\varepsilon}(S_j, S_i, S_l, Q)$. Then the predicate can be answered by looking at the relative position of t_q with respect to t_i and t_j on ∂B_{ijk} (or the relative position of t'_q with respect to t'_i and t'_j on ∂B_{jil}).

More precisely:

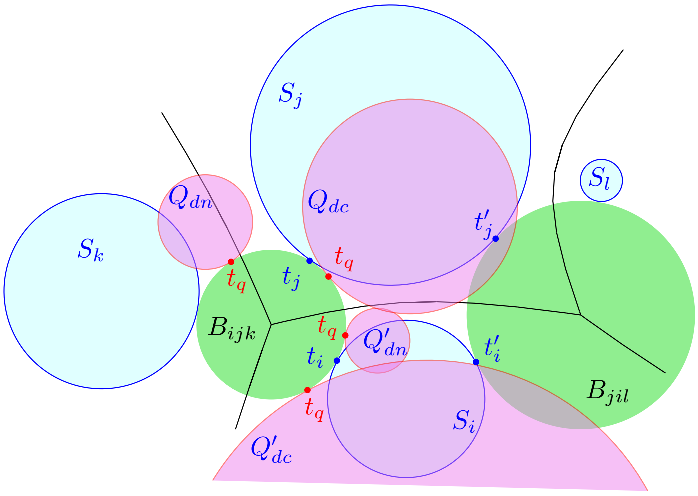

Figure 10: The different degenerate possible configurations for the EdgeConflict predicate.

- if $\mathsf{VertexConflict}^\varepsilon(S_i, S_j, S_k, Q) = \mathsf{VertexConflict}^\varepsilon(S_j, S_i, S_l, Q) =$ "no conflict" and $\mathsf{VertexConflict}(S_i, S_j, S_k, Q) =$ "degenerate", then the answer is "no conflict" if $t_i t_j t_q$ is ccw (Q_{dn} on Figure 10), "conflict interior" otherwise (Q'_{dn} on Figure 10);
- if $\mathsf{VertexConflict}^\varepsilon(S_i, S_j, S_k, Q) = \mathsf{VertexConflict}^\varepsilon(S_j, S_i, S_l, Q) =$ "conflict" and $\mathsf{VertexConflict}(S_i, S_j, S_k, Q) =$ "degenerate", then the answer is "conflict both" if $t_i t_j t_q$ is ccw (Q_{dc} on Figure 10), "conflict entire edge" otherwise (Q'_{dc} on Figure 10);
- if $\mathsf{VertexConflict}(S_i, S_j, S_k, Q) = \mathsf{VertexConflict}^\varepsilon(S_j, S_i, S_l, Q) =$ "no conflict" and $\mathsf{VertexConflict}(S_j, S_i, S_l, Q) =$ "degenerate", then the answer is "no conflict" if $t'_j t'_i t'_q$ is ccw, "conflict interior" otherwise;
- if $\mathsf{VertexConflict}(S_i, S_j, S_k, Q) = \mathsf{VertexConflict}^\varepsilon(S_j, S_i, S_l, Q) =$ "conflict" and $\mathsf{VertexConflict}(S_j, S_i, S_l, Q) =$ "degenerate", then the answer is "conflict both" if $t'_j t'_i t_q$ is ccw, "conflict entire edge" otherwise.

The main observation from the above analysis is that the max-weight qualitative perturbation scheme described in Section 3.3.2 not only resolves the VertexConflict predicate, but also the EdgeConflict predicate (although not described in this paper, our perturbation scheme resolves, in fact, all degeneracies of all predicates used in [9] for the computation of the 2D Apollonius diagram). To resolve the EdgeConflict predicate, we need only evaluate $\mathsf{Orientation}(t_i, t_j, t_q)$ and/or $\mathsf{Orientation}(t'_j, t'_i, t'_q)$. As described in the previous section, this predicate is of algebraic degree 8, and thus does not increase the algebraic degree of the EdgeConflict predicate. Furthermore, with a careful implementation, it is possible to keep track of the intermediate results of the evaluation of the $\mathsf{VertexConflict}^\varepsilon$ predicate and resolve the EdgeConflict predicate using these intermediate results in a purely combinatorial manner.

As a final note, the above procedure for the evaluation of the EdgeConflict predicate works as is even when the Apollonius edge e is of zero length. This is the case when the Apollonius vertices v_{ijk} and v_{jil} coincide, or, equivalently, when the four sites S_i, S_l, S_j and S_k are all tangent (and in that ccw order) to the same Apollonius circle. The only difference with respect to the non-zero-length Apollonius edge case are the possible outcomes: the EdgeConflict predicate will never return "conflict interior" nor "conflict both"; this is, however, automatically handled by the procedure described above, that is without the need to handle any additional special cases.

3.4 Perturbing spheres for the 3D Apollonius diagram

In three dimensions, an Apollonius vertex v_{ijkl} is defined by four sites S_i, S_j, S_k, and S_l, while the precidate $\mathsf{VertexConflict}(S_i, S_j, S_k, S_l, Q)$ tests if after adding a fifth site $Q = S_q$, v_{ijkl} remains a valid Apollonius vertex or not. Let B_{ijkl} denote the ball tangent to S_i, S_j, S_k, and S_l whose tangency points t_i, t_j, t_k, and t_l forms a positively oriented tetrahedron $t_it_jt_kt_l$.

The predicate in general position can be solved in different ways. One way is to do like Boissonnat and Delage [2]. Another way is to use inversion like in the 2D case by Karavelas and Emiris [9], and arrive at an alternative expression. In degenerate configuration, Q and B_{ijkl} are tangent at t_q and we can obtain, as a side product, the orientation of the tetrahedra formed by 4 of the 5 tangency points t_i, t_j, t_k, t_l, and t_q.

As for the 2D case, we apply the max-weight qualitive perturbation scheme. If the predicate is degenerate the effect of the perturbation is that the weight of the site with largest index increases, and thus intersects the ball tangent to the four other sites. In the neighborhood of the center of B_{ijkl} the Apollonius diagram of S_i, S_j, S_k, S_l, and Q has the same combinatorial structure as the Voronoi diagram of t_i, t_j, t_k, t_l, and t_q. We, thus, get an equivalent formulation for the predicate: given five co-spherical points t_i, t_j, t_k, t_l, and t_q, does the tetrahedron $t_it_jt_kt_l$ remain in the Delaunay triangulation when the point of the largest index is moved inside the ball. Notice that it implies that the point with largest index is linked to all other points in this Delaunay triangulation.

Similarly to the two-dimensional case, we can conclude that Q is in conflict with B_{ijkl} if $q > i, j, k, l$. We can also take care of the cases where t_q is equal to one of the four points t_i, t_j, t_k, and t_l. Otherwise, we rename the indices so that i is the largest one. The definition of B_{ijkl} says that tetrahedron $t_it_jt_kt_l$ is positively oriented. Notice that this can be true in two ways: either the tetrahedron is really positively oriented, or it is flat, as the limit of a positively oriented tetrahedron when $\varepsilon_i \to 0^+$.

If $t_qt_jt_kt_l$ is positively oriented, t_q and t_i are on the same side of $t_jt_kt_l$, which is a convex hull facet. Since the 3D Apollonius graph is "star shaped" from S_i, S_i is linked to $S_jS_kS_l$ to create the tetrahedron $S_iS_jS_kS_l$, and thus there is no conflict for Q.

If $t_qt_jt_kt_l$ is negatively oriented, t_q and t_i are on opposite sides of $t_jt_kt_l$, which implies that $S_iS_jS_kS_l$ ceases to be a facet of the Apollonius graph. Thus, the tetrahedron $S_iS_jS_kS_l$ disappears and Q is in conflict.

If $t_qt_jt_kt_l$ is flat, the question reduces to determining the orientation of $t_q^\varepsilon t_j^\varepsilon t_k^\varepsilon t_l^\varepsilon$, which are the points of tangency of S_q, S_j, S_k, S_l with B_{ijkl}^ε after the perturbation of S_i by ε_i. This orientation will be non-degenerate except in two very special cases where the centers of Q, S_j, S_k, and S_l are either co-circular or collinear.

We first address the case where S_j, S_k, and S_l are neither co-circular nor collinear. Let us assume that l is smaller than j and k, which means that $w_l \leq w_j, w_k$. By subtracting w_l from all weights, we can consider that S_l has zero weight, and then perform an inversion with pole s_l (see Figure 11). Let Z_i, Z_j, Z_k, and Z_q be the images of sites S_i, S_j, S_k, and Q, and ω_i, ω_j, ω_k, and ω_q be the images of t_i, t_j, t_k, and t_q under inversion, and denote by z_i, z_j, z_k, and z_q the centers of Z_i, Z_j, Z_k, and Z_q. Since $t_qt_jt_kt_l$ is a flat tetrahedron, the four points t_q, t_j, t_k, t_l are co-circular and thus $\omega_q, \omega_j, \omega_k$ are collinear. Their supporting line lies in the *unique* plane Π_{ijk} that is commonly tangent to all three sites Z_q, Z_j, and Z_k. The uniqueness follows from the fact that the z_q, z_j, and z_k are not collinear, since s_q, s_j, s_k, and $s_l = t_l$ have been assumed to be neither co-circular nor collinear. The planes tangent to both Z_j and Z_k are tangents to

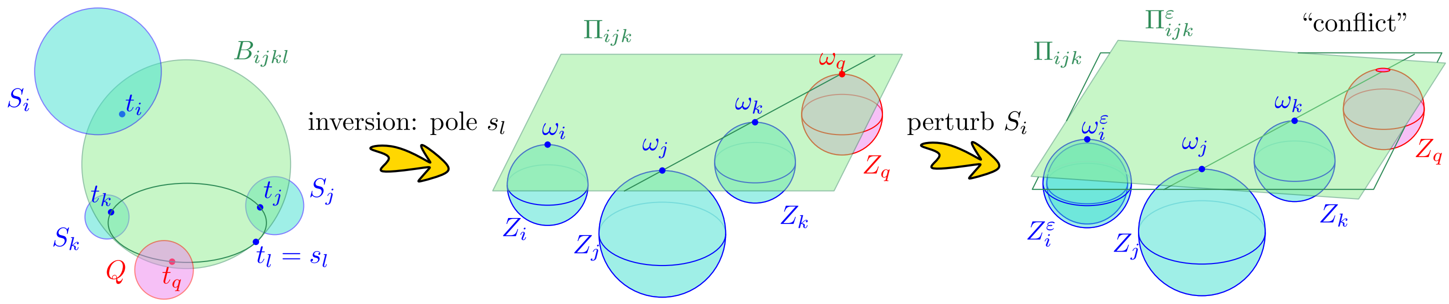

Figure 11: The case, for the VertexConflict predicate, where the tetrahedron $t_q t_j t_k t_l$ is flat.

the cone C, whose axis is the line through z_j and z_k. When we perturb S_i to S_i^{ε}, the plane Π_{ijk} moves a bit, in the set of planes tangent to C, to become Π^{ε}_{ijk}.

Consider first the case $w_q \geq w_l$, which implies that the weight of Z_q is non-negative. Since z_q, z_j, and z_k are not collinear and Z_q is tangent to C at ω_q, Z_q either properly intersects C or is inside C. If Z_q is (tangent to and) inside C, then, for all values of ε_i, Π^{ε}_{ijk} does not intersect Z_q, and the result of the perturbed predicate is "no conflict", otherwise, for all values of ε_i, Π^{ε}_{ijk} intersects Z_q, and the result of the perturbed predicate is "conflict". The way to evaluate the VertexConflict$^{\varepsilon}$ predicate, in this case, is by determining the value of Orientation(z_q, z_j, z_k), where this orientation is seen as a two-dimensional orientation in the plane that is perpendicular to Π_{ijk} and passes through z_j and z_k. If $w_j = w_k = w_l$, the cone C degenerates to the line through z_j and z_k. In this case we have $w_q > w_l$ (since, otherwise, s_j, s_k, s_l and s_q would be co-circular), and thus VertexConflict$^{\varepsilon}$ returns "conflict". If at least two of w_j, w_k, w_l differ, then at least one of ω_j and ω_k differs from z_j and z_k, respectively. Denoting by $\omega_\star$ a/the point of tangency that differs from the corresponding center, it suffices to determine if Orientation$(z_q, z_j, z_k) =$ Orientation$(\omega_\star, z_j, z_k)$, in which case the VertexConflict$^{\varepsilon}$ predicate returns "no conflict", otherwise "conflict" is returned.

Finally, notice that when $w_q < w_l$, the site Z_q has negative weight. In this case, the sphere Z_q^- will properly intersect the plane Π^{ε}_{ijk} for all values of ε_i, which implies that S_q does not intersect B^{ε}_{ijk}. Hence, in this case, the result of the perturbed predicate is "no conflict".

In the very degenerate cases where s_j, s_k, s_l, and s_q are co-circular or collinear, the unique edge of the (degenerate) Apollonius diagram of S_j, S_k, S_l, and Q is a circle or a line. In these cases, the position of S_i has no real influence on the diagram of S_i, S_j, S_k, S_l, and Q, and we need to perturb the second most perturbed site S_j or S_k or Q to remove the degeneracy. The resolution of the degeneracy is similar to the 2D case: first perform a positive permutation of j, k, l to ensure that $j > k, l$. If $q > j$ then Q will be in conflict, otherwise, if $q < j$ then Q will be in conflict if and only if $t_j t_k t_l$ and $t_q t_k t_l$ have different orientations. In this very special configuration, the orientations of the tangency points are the same with the orientations of the site centers, and, hence, they only involve the basic orientation predicate.

Following the above analysis, $\mathsf{VertexConflict}^{\varepsilon}(S_i, S_j, S_k, S_l, Q)$ can be evaluated as follows:

1. if $\mathsf{VertexConflict}(S_i, S_j, S_k, S_l, Q) \neq$ "degenerate", then return $\mathsf{VertexConflict}(S_i, S_j, S_k, S_l, Q)$;
2. if $q > i, j, k, l$, then return "conflict";
3. ensure that $i > j, k > l$ by a positive permutation of (i, j, k, l);
4. if $t_q = t_i$, then return "no conflict";
5. if $t_q = t_j$, then { if $q > j$ return "conflict", else return "no conflict"};
6. if $t_q = t_k$, then { if $q > k$ return "conflict", else return "no conflict"};
7. if $t_q = t_l$, then { if $q > l$ return "conflict", else return "no conflict"};
8. if $t_q t_j t_k t_l$ is positively oriented, then return "no conflict";
9. if $t_q t_j t_k t_l$ is negatively oriented, then return "conflict";
10. if s_j, s_k, s_l, s_q are neither collinear nor co-circular, then:
    i. if $w_q < w_l$ return "no conflict";
    ii. if $w_j = w_k = w_l$ return "conflict";
    iii. compute a/the tangency point $\omega_\star$;
    if z_q, z_j, z_k and $\omega_\star, z_j, z_k$ have the same 2D orientation
    return "no conflict", else return "conflict";
11. ensure that $i > j > k, l$ by a positive permutation of (j, k, l);
12. if $q > j$ return "conflict";
13. if $t_j t_k t_l x$ and $t_q t_k t_l x$ have the same orientation (for any point $x \notin plane(t_j t_k t_l)$)
    return "no conflict", else return "conflict".

Steps 8 and 9 of the above algorithm rely on the auxiliary predicate $\mathsf{Orientation}(t_q, t_j, t_k, t_l)$ that, given five sites, computes the orientation of the four tangency points on the common tangent sphere to the fifth site. Since the tetrahedron $t_i t_j t_k t_l$ is, by definition positively oriented, $\mathsf{Orientation}(t_q, t_j, t_k, t_l)$ will be positive if and only if t_q lies on the upper *half* of B_{ijkl} as t_i, where the two halves of B_{ijkl} are delimited by the circle through t_j, t_k and t_l. We first reduce all the weights by w_l. If $w_j = w_k = w_l (= 0)$, computing the orientation of t_q, t_j, t_k, t_l amounts to evaluating the orientation of t_q, s_j, s_k, s_l. If w_j, w_k and w_l are not all equal, we consider again the inversion transformation with s_l as the pole. Then t_q lies on the same half of B_{ijkl} as t_i if and only if ω_q lies on the same half-plane of Π_{ijk}, with respect to the line through ω_j and ω_k, with ω_i. The equality of these 2D orientation tests is equivalent to testing the result of $\mathsf{Orientation}(z_q, z_j, z_k, \omega_\star)$; if $\mathsf{Orientation}(z_q, z_j, z_k, \omega_\star) =$ "ccw" return "no conflict", otherwise return "conflict". For Step 10 we first need to test if the points s_j, s_k, s_l and s_q are either collinear or co-circular. The possible collinearity can easily be tested via the cross-products $(s_l - s_j) \times (s_k - s_j)$ and $(s_q - s_j) \times (s_k - s_j)$; if both are the zero vector, the four points are collinear. Co-circularity in the original space corresponds to collinearity in the inverted space (where the pole of inversion is s_l); to test if s_j, s_k, s_l and s_q are co-circular we simply need to test if z_j, z_k, z_q are collinear, which amounts to computing the cross-product $(z_q - z_j) \times (z_k - z_j)$. For the 2D orientations of z_q, z_j, z_k and $\omega_\star, z_j, z_k$, we need only choose a point $x \notin plane(z_q z_j z_k)$ and determine if the tetrahedra $z_q z_j z_k x$ and $\omega_\star z_j z_k x$ have the same orientation. Finally, as discussed above, the orientations of $t_j t_k t_l x$ and $t_q t_k t_l x$ in Step 13 are the same as those of $s_j s_k s_l x$ and $s_q s_k s_l x$, which are standard 3D orientation tests.

4 Predicates for circular arcs arrangements

To address the problem of computing arrangements of circular arcs by sweep-line algorithms, it is necessary to compare abscissae of endpoints of circular arcs. This is precisely the predicate studied in a previous paper [4]. The arc endpoints are described as intersections of two circles, which leads us to consider the arrangement of all circles supporting arcs or defining their endpoints. Degeneracies occur if several vertices of the arrangement have the same abscissa or if

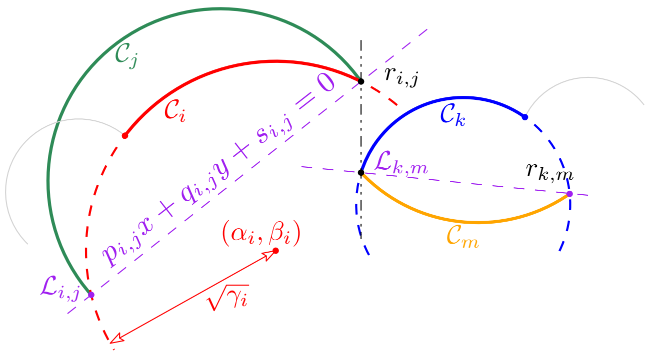

Figure 12: x-comparison of endpoints: a degenerate case where $r_{i,j} = l_{k,m}$

more than two circles meet at a common point. For arrangements with a lot of degeneracies, it may be interesting to design an algorithm that directly handles special cases, while in other contexts where degeneracies are occasional, it would be preferable to keep the algorithm simple and handle degeneracies through a perturbation scheme.

An endpoint $z_{\nu,\mu}$, defined as an intersection of two circles $\mathcal{C}_\nu$ and $\mathcal{C}_\mu$, is determined by the centers (α_ν, β_ν) and (α_μ, β_μ) of the circles, their squared radii γ_ν and γ_μ, and a Boolean encoding whether $z_{\nu,\mu}$ is the left-most $l_{\nu,\mu}$ or right-most intersection point $r_{\nu,\mu}$ (if they have the same abscissa, $r_{\nu,\mu}$ is the highest and $l_{\nu,\mu}$ the lowest intersection point) (see Figure 12).

4.1 Algebraic formulation

Intersection points of $\mathcal{C}_\nu$ and $\mathcal{C}_\mu$ can also be seen as intersections between $\mathcal{C}_\nu$ and the line $\mathcal{L}_{\nu,\mu}$ whose equation $p_{\nu,\mu}x + q_{\nu,\mu}y + s_{\nu,\mu} = 0$ is obtained by subtracting the equations of $\mathcal{C}_\nu$ and $\mathcal{C}_\mu$: $p_{\nu,\mu} = 2(\alpha_\nu - \alpha_\mu)$, $q_{\nu,\mu} = 2(\beta_\nu - \beta_\mu)$, and $s_{\nu,\mu} = \gamma_\nu - \gamma_\mu - \alpha_\nu^2 - \beta_\nu^2 + \alpha_\mu^2 + \beta_\mu^2$.

The predicate x-$\mathsf{compare}(\mathcal{C}_i, \mathcal{C}_j, \mathcal{C}_k, \mathcal{C}_l)$ compares the abscissae of two arc endpoints $z_{i,j}$ defined by $\mathcal{C}_i$ and $\mathcal{C}_j$, $i \neq j$, and $z_{k,m}$ defined by $\mathcal{C}_k$ and $\mathcal{C}_l, k \neq l$. The most critical evaluation is the sign of the following degree 12 polynomial [4]:

$$(A_{i,j}\, C_{k,m} - A_{k,m}\, C_{i,j})^2 - 4(A_{i,j}\, B_{k,m} - A_{k,m}\, B_{i,j})(B_{i,j}\, C_{k,m} - B_{k,m}\, C_{i,j})$$

where:
$$\begin{aligned} A_{\nu,\mu} &= p_{\nu,\mu}^2 + q_{\nu,\mu}^2, \\ B_{\nu,\mu} &= q_{\nu,\mu}^2\, \alpha_\nu - p_{\nu,\mu}\, s_{\nu,\mu} - p_{\nu,\mu}\, q_{\nu,\mu}\, \beta_\nu, \\ C_{\nu,\mu} &= s_{\nu,\mu}^2 + 2\, q_{\nu,\mu}\, s_{\nu,\mu}\, \beta_\nu + q_{\nu,\mu}^2\, \alpha_\nu^2 + q_{\nu,\mu}^2\, \beta_\nu^2 - q_{\nu,\mu}^2\, \gamma_\nu. \end{aligned}$$

As for the problems we have considered above, introducing an algebraic symbolic perturbation yields a quite complicated polynomial in ε.

4.2 Qualitative symbolic perturbation

Qualitative symbolic perturbation. We construct a sequence of $n+1$ successive perturbations for an input $\boldsymbol{\mathcal{C}}$ of n circles $\mathcal{C}_\nu$, $0 \leq \nu < n$.

As a first perturbation, $\pi_n(\boldsymbol{\mathcal{C}}, \varepsilon_n)$ is a rotation centered at the origin and with angle ε_n. This perturbation handles the cases where $z_{i,j}$ and $z_{l,m}$ are different points with the same abscissa.

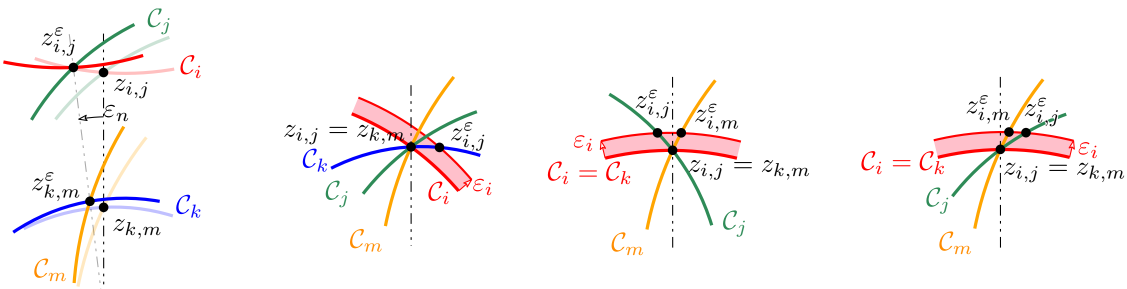

Figure 13: Perturbing $z_{i,j}$ and $z_{k,m}$

In other words, it just relies on the lexicographic ordering to break ties in x-comparisons by using y-comparisons(see Figure 13-a). The rotation by a small angle has the same effect as a shear transform. A shear transform is often preferred in the literature because it is a rational transformation. We prefer a rotation because it transforms circles into circles, which is preferable in order to apply the remaining perturbations in the sequence. On top of that, we are not interested in the algebraic formulation of the transformation, since we look at the limit geometrically and not algebraically. The other perturbations in the sequence consist in inflating the circles: $\pi_\nu(\boldsymbol{\mathcal{C}}, \varepsilon_\nu)$ replaces γ_ν by $\gamma_\nu^\varepsilon = \gamma_\nu + \varepsilon_\nu$, $\varepsilon_\nu \geq 0$. As for the Apollonius diagram, we consider the circles by decreasing radii to get rid of pairs of tangent circles: if two circles are tangent, the perturbation inflates the largest one more, making the intersection point disappear.

Wlog we may assume that $i \geq j, k, m$, i.e., $\mathcal{C}_i$ is the most perturbed circle.

If $i \notin \{k, m\}$, then $z_{i,j}$ moves while $z_{l,m}$ stays fixed. After determining the vertical order of $\mathcal{C}_i$ and $\mathcal{C}_j$ at the right of $z_{i,j}$, and whether $z_{i,j}$ is on the top or bottom part of $\mathcal{C}_i$ using the auxiliary predicates described below, it is easy to decide if $z_{i,j}^\varepsilon$ moves left or right when $\varepsilon_i > 0$ (see Figure 13-b).

If $i \in \{k, m\}$, assume, wlog, that $i = k$. If $z_{i,j}^\varepsilon$ and $z_{i,m}^\varepsilon$ are perturbed in opposite x-directions, it is easy to decide which one is the left-most (see Figure 13-c). Otherwise, we determine the vertical order of the three circles $\mathcal{C}_i$, $\mathcal{C}_j$, and $\mathcal{C}_m$ at the right of $z_{i,j} = z_{i,m}$; we know that $\mathcal{C}_i$ is either the top-most or the bottom-most circle in this vertical ordering. The point lying on the closest arc to $\mathcal{C}_i$ is more perturbed than the other, and the auxiliary predicates allow us to decide which of $z_{i,j}^\varepsilon$ and $z_{i,m}^\varepsilon$ is left of the other (see Figure 13-d).

Auxiliary predicates. A circle can be split in four parts: top-right, top-left, bottom-left, and bottom-right at its points with horizontal or vertical tangents. Knowing if a point $z_{i,j}$ is on the left or right part of $\mathcal{C}_i$ can be evaluated by x-$\mathsf{compare}(\mathcal{C}_i, \mathcal{C}_j, \mathcal{C}_i, \mathcal{C}_m)$ such that $\mathcal{L}_{i,m}$ has equation $x - \alpha_i = 0$. Discriminating between the top and bottom parts is done in the same way, by exchanging the roles of the x- and y-coordinates. Another predicate consists in deciding if $\mathcal{C}_i$ is above or below $\mathcal{C}_j$ at the right of $r_{i,j}$; this can be done by elementary geometric computations.

5 Conclusion

In this paper, a new framework for dealing with geometric degeneracies has been proposed: the qualitative symbolic perturbation. Conversely to usual approaches for symbolic perturbation, the new framework does not rely on a particular algebraic description of the predicate, but rather directly on its geometric description.

A qualitative symbolic perturbation scheme consists of a sequence of perturbations, but given a specific predicate only a few of these perturbations are really *active*. The number of active perturbations used to resolve a specific predicate depends on the problem at hand. For the 2D Apollonius diagram perturbing one site always suffices. In its 3D counterpart we may need to perturb two sites, whereas in the case of circular arcs we may need perform a rotation (perturb the axes) and perturb up to one supporting circle per predicate. Minimizing the number of active perturbations is not necessarily desirable, since it might result in a more complicated predicate evaluation analysis (for example trying to resolve degeneracies for the trapezoidal map of circular arcs with a single active perturbation seems much more complicated).

Besides the number of active perturbations, another important issue is the ordering of the perturbations: for the Apollonius diagram we consider sites by descreasing weight, whereas for the trapezoidal map of circular arcs we first consider a (global) rotation and then the circles by means of descreasing radius. Different perturbation sequences than the ones described in this paper are definitely possible; the analysis, however, can become unnecessarily more complicated.

Our qualitative symbolic perturbation framework, and in particular the schemes described in this paper, can also be applied to a variety of other problems, such as the 2D Voronoi diagram of disjoint convex objects under any L_p metric, as well as the Euclidean Voronoi diagram of certain disjoint convex objects in 3D (the objects can be, for example, non-intersecting lines, line segments or rays). It suffices to replace a site S_i with its Minkowski sum with a ball of radius ε_i, and then consider the limits $\varepsilon_i \to 0$, for an appropriately defined ordering of the sites. We would like to extend the applicability of our framework to the evaluation of the predicates for other geometric structures as well, such as the predicates appearing in the computation of lines tangent to four given lines in 3D [3, 5].

As a parallel goal, we plan to implement the qualitative perturbation schemes for the problems presented in this paper. In fact, the implementation of the max-weight perturbation scheme inside the `Apollonius_graph_2` package of CGAL [10] is already under way, and is expected to become part of the package in the near future.

References

[1] Pierre Alliez, Olivier Devillers, and Jack Snoeyink. Removing degeneracies by perturbing the problem or the world. *Reliable Computing*, 6:61–79, 2000.
http://dx.doi.org/10.1023/A:1009942427413.

[2] Jean-Daniel Boissonnat and Christophe Delage. Convex hull and Voronoi diagram of additively weighted points. In G. S. Brodal and S. Leonardi, editors, *Proceedings of 13th Annual European Symposium on Algorithms (ESA 2005)*, volume 3669 of *LNCS*, pages 367–378. Springer, 2005.
http://dx.doi.org/10.1007/11561071_34.

[3] H. Brönnimann, O. Devillers, Vida Dujmović, H. Everett, M. Glisse, X. Goaoc, S. Lazard, H.-S. Na, and S. Whitesides. Lines and free line segments tangent to arbitrary three-dimensional convex polyhedra. *SIAM Journal on Computing*, 37:522–551, 2007.
http://dx.doi.org/10.1137/S0097539705447116.

[4] Olivier Devillers, Alexandra Fronville, Bernard Mourrain, and Monique Teillaud. Algebraic methods and arithmetic filtering for exact predicates on circle arcs. *Computational Geometry: Theory and Applications*, 22:119–142, 2002.
http://dx.doi.org/10.1016/S0925-7721(01)00050-5.

[5] Olivier Devillers, Marc Glisse, and Sylvain Lazard. Predicates for line transversals to lines and line segments in three-dimensional space. In *Proc. 24th Annual Symposium on Computational Geometry*, pages 174–181, 2008.
http://doi.acm.org/10.1145/1377676.1377704.

[6] Olivier Devillers and Monique Teillaud. Perturbations for Delaunay and weighted Delaunay 3D triangulations. *Computational Geometry: Theory and Applications*, 44:160–168, 2011.
http://dx.doi.org/10.1016/j.comgeo.2010.09.010.

[7] H. Edelsbrunner and E. P. Mücke. Simulation of simplicity: A technique to cope with degenerate cases in geometric algorithms. *ACM Trans. Graph.*, 9(1):66–104, 1990.
http://dx.doi.org/10.1145/77635.77639.

[8] I. Emiris and J. Canny. A general approach to removing degeneracies. *SIAM J. Comput.*, 24:650–664, 1995.
http://dx.doi.org/10.1137/S0097539792235918.

[9] Ioannis Z. Emiris and Menelaos I. Karavelas. The predicates of the Apollonius diagram: algorithmic analysis and implementation. *Computational Geometry: Theory and Applications*, 33(1-2):18–57, January 2006. Special Issue on Robust Geometric Algorithms and their Implementations.
http://dx.doi.org/10.1016/j.comgeo.2004.02.006.

[10] Menelaos Karavelas and Mariette Yvinec. 2D Apollonius graphs (Delaunay graphs of disks). In *CGAL User and Reference Manual*. CGAL Editorial Board, 4.1 edition, 2012.
http://www.cgal.org/Manual/latest/doc_html/cgal_manual/packages.html#Pkg:ApolloniusGraph2.

[11] R. Klein, K. Mehlhorn, and S. Meiser. Randomized incremental construction of abstract Voronoi diagrams. *Comput. Geom.: Theory & Appl.*, 3(3):157–184, 1993.
http://dx.doi.org/10.1016/0925-7721(93)90033-3.

[12] R. Seidel. The nature and meaning of perturbations in geometric computing. *Discrete Comput. Geom.*, 19:1–17, 1998.
http://dx.doi.org/10.1007/PL00009330.

[13] C. K. Yap. A geometric consistency theorem for a symbolic perturbation scheme. *J. Comput. Syst. Sci.*, 40(1):2–18, 1990.
http://dx.doi.org/10.1016/0022-0000(90)90016-E.

[14] C. K. Yap. Symbolic treatment of geometric degeneracies. *J. Symbolic Comput.*, 10:349–370, 1990.
http://dx.doi.org/10.1016/S0747-7171(08)80069-7.

A Analysis of the predicates for the 2D Apollonius diagram

In this section we introduce a slightly heavier, yet more general notation. In particular, we introduce the following determinant shorthands:

$$D^s_{\mu\nu} = \begin{vmatrix} s_\mu & 1 \\ s_\nu & 1 \end{vmatrix} = s_\mu - s_\nu, \qquad D^{st}_{\mu\nu} = \begin{vmatrix} s_\mu & t_\mu \\ s_\nu & t_\nu \end{vmatrix}, \qquad D^{st}_{\mu\nu\lambda} = \begin{vmatrix} s_\mu & t_\mu & 1 \\ s_\nu & t_\nu & 1 \\ s_\lambda & t_\lambda & 1 \end{vmatrix},$$

with $s, t \in \{x, y, u, v, \rho\}$ and $\mu, \nu, \lambda \in \{j, k, q\}$, and

$$E^s_{\mu\nu} = \begin{vmatrix} s^*_\mu & p^*_\mu \\ s^*_\nu & p^*_\nu \end{vmatrix}, \qquad E^{st}_{\mu\nu} = \begin{vmatrix} s^*_\mu & t^*_\mu \\ s^*_\nu & t^*_\nu \end{vmatrix}, \qquad E^{st}_{\mu\nu\lambda} = \begin{vmatrix} s^*_\mu & t^*_\mu & p^*_\mu \\ s^*_\nu & t^*_\nu & p^*_\nu \\ s^*_\lambda & t^*_\lambda & p^*_\lambda \end{vmatrix},$$

with $s, t \in \{x, y, w\}$ and $\mu, \nu, \lambda \in \{j, k, q\}$, and

$$x^*_\nu = x_\nu - x_i, \quad y^*_\nu = y_\nu - y_i, \quad w^*_\nu = w_\nu - w_i, \quad p^*_\nu = (x^*_\nu)^2 + (y^*_\nu)^2 - (w^*_\nu)^2, \quad \nu \in \{j, k, q\}.$$

A.1 The algebraic degree of the VertexConflict predicate

Recall from Section 3.3.1 that the VertexConflict predicate can be resolved by determining the sign of the quantity:

$$I := E^{xw}_{jkq} E^x_{jk} + E^{yw}_{jkq} E^y_{jk} + E^{xy}_{jkq} \sqrt{\Delta}, \qquad \Delta = (E^x_{jk})^2 + (E^y_{jk})^2 - (E^w_{jk})^2.$$

This is a quantity of the form $X_0 + X_1\sqrt{Y}$, which means that we might need to compute the sign of the quantity $Z = X_0^2 - X_1^2 Y$. In our case Z can be factorized as described below:

$$\begin{aligned}
Z &= (E^{xw}_{jkq} E^x_{jk} + E^{yw}_{jkq} E^y_{jk})^2 - (E^{xy}_{jkq})^2((E^x_{jk})^2 + (E^y_{jk})^2 - (E^w_{jk})^2) \\
&= (E^{xw}_{jkq} E^x_{jk})^2 + (E^{yw}_{jkq} E^y_{jk})^2 + 2E^x_{jk} E^y_{jk} E^{yw}_{jkq} E^{xw}_{jkq} - (E^{xy}_{jkq})^2[(E^x_{jk})^2 + (E^y_{jk})^2] + (E^{xy}_{jkq} E^w_{jk})^2 \\
&= (E^{xw}_{jkq})^2 (E^x_{jk})^2 + (E^{yw}_{jkq})^2 (E^y_{jk})^2 + (E^{yw}_{jkq})^2 (E^x_{jk})^2 + (E^{xw}_{jkq})^2 (E^y_{jk})^2 - (E^{xy}_{jkq})^2[(E^x_{jk})^2 + (E^y_{jk})^2] \\
&\quad - (E^{yw}_{jkq} E^x_{jk})^2 - (E^{xw}_{jkq} E^y_{jk})^2 + 2E^x_{jk} E^y_{jk} E^{yw}_{jkq} E^{xw}_{jkq} + (E^{xy}_{jkq} E^w_{jk})^2 \\
&= (E^{xw}_{jkq})^2[(E^x_{jk})^2 + (E^y_{jk})^2] + (E^{yw}_{jkq})^2[(E^x_{jk})^2 + (E^y_{jk})^2] - (E^{xy}_{jkq})^2[(E^x_{jk})^2 + (E^y_{jk})^2] \\
&\quad + (E^{xy}_{jkq} E^w_{jk})^2 - (E^{xw}_{jkq} E^y_{jk} - E^{yw}_{jkq} E^x_{jk})^2 \\
&= [(E^{xw}_{jkq})^2 + (E^{yw}_{jkq})^2 - (E^{xy}_{jkq})^2][(E^x_{jk})^2 + (E^y_{jk})^2] + (E^{xy}_{jkq} E^w_{jk})^2 - (E^{xw}_{jkq} E^y_{jk} - E^{yw}_{jkq} E^x_{jk})^2.
\end{aligned}$$

It easy to verify that the following identities hold:

$$\begin{aligned}
p^*_j E^{xw}_{jkq} &= E^x_{Qj} E^w_{jk} - E^x_{jk} E^w_{Qj}, \\
p^*_j E^{yw}_{jkq} &= E^y_{Qj} E^w_{jk} - E^y_{jk} E^w_{Qj}, \text{ and} \\
p^*_j E^{xy}_{jkq} &= E^x_{jk} E^y_{Qj} - E^y_{jk} E^x_{Qj},
\end{aligned}$$

which implies that

$$\begin{aligned}
p^*_j(E^{xw}_{jkq} E^y_{jk} - E^{yw}_{jkq} E^x_{jk}) &= (E^x_{Qj} E^w_{jk} - E^x_{jk} E^w_{Qj}) E^y_{jk} - (E^y_{Qj} E^w_{jk} - E^y_{jk} E^w_{Qj}) E^x_{jk} \\
&= E^x_{Qj} E^y_{jk} E^w_{jk} - E^x_{jk} E^y_{jk} E^w_{Qj} - E^x_{jk} E^y_{Qj} E^w_{jk} + E^x_{jk} E^y_{jk} E^w_{Qj} \\
&= E^x_{Qj} E^y_{jk} E^w_{jk} - E^x_{jk} E^y_{Qj} E^w_{jk} \\
&= (E^x_{Qj} E^y_{jk} - E^x_{jk} E^y_{Qj}) E^w_{jk} \\
&= -p^*_j E^{xy}_{jkq} E^w_{jk},
\end{aligned}$$

Since $p_j^* \neq 0$, we have

$$E^{xw}_{jkq} E^{y}_{jk} - E^{yw}_{jkq} E^{x}_{jk} = -E^{xy}_{jkq} E^{w}_{jk}. \tag{4}$$

Hence $(E^{xy}_{jkq} E^{w}_{jk})^2 - (E^{xw}_{jkq} E^{y}_{jk} - E^{yw}_{jkq} E^{x}_{jk})^2 = 0$, and Z simplifies to:

$$Z = [(E^{x}_{jk})^2 + (E^{y}_{jk})^2][(E^{xw}_{jkq})^2 + (E^{yw}_{jkq})^2 - (E^{xy}_{jkq})^2]. \tag{5}$$

Clearly, the two factors of Z are of degree 6 and 8, respectively. Moreover, as discussed in Section 3.3.1, when we evaluate Z it cannot be the case that $E^{x}_{jk} = E^{y}_{jk} = 0$ (since then we would have resolved the sign of I without resorting to Z); as a result the degree of the VertexConflict predicate is 8.

A.2 The auxiliary predicate for VertexConflict$^{\varepsilon}$ using inversion

In this section we assume that we are given three sites $S_\mu = (x_\mu, y_\mu, w_\mu)$, $\mu \in \{i, j, k\}$ that define an Apollonius vertex, and a query site $Q = (x_q, y_q, w_q)$ such that VertexConflict$(S_i, S_j, S_k, Q) =$ "degenerate" (see Figure 7). We are going to compute the VertexConflict$^{\varepsilon}(S_i, S_j, S_k, Q)$ predicate in the first manner described in Section 3.3.3 (cf. page 12).

Following the analysis in [9], the equation of the line L in the plane of inversion is $au+bv+c = 0$, where:

$$\begin{aligned}
a &= \frac{D^u_{jk} D^w_{jk} + D^v_{jk}\sqrt{\Gamma}}{(D^u_{jk})^2 + (D^u_{jk})^2} &&= \frac{E^x_{jk} E^w_{jk} + E^y_{jk}\sqrt{\Delta}}{(E^x_{jk})^2 + (E^y_{jk})^2},\\
b &= \frac{D^v_{jk} D^w_{jk} - D^u_{jk}\sqrt{\Gamma}}{(D^u_{jk})^2 + (D^u_{jk})^2} &&= \frac{E^y_{jk} E^w_{jk} - E^x_{jk}\sqrt{\Delta}}{(E^x_{jk})^2 + (E^y_{jk})^2},\\
c &= \frac{D^u_{jk} D^{u\rho}_{jkq} + D^v_{jk} D^{v\rho}_{jkq} + D^{uv}_{jkq}\sqrt{\Gamma}}{(D^u_{jk})^2 + (D^u_{jk})^2} &&= \frac{E^x_{jk} E^{xw}_{jkq} + E^y_{jk} E^{yw}_{jkq} + E^{xy}_{jkq}\sqrt{\Delta}}{(E^x_{jk})^2 + (E^y_{jk})^2}.
\end{aligned}$$

where $\Gamma = (D^u_{jk})^2 + (D^v_{jk})^2 - (D^\rho_{jk})^2$. Therefore, the line $L^{\perp}_{\nu}$ that is perpendicular to L and passes through c_ν has equation (written in the coordinate system of the plane of inversion):

$$\beta(u - u_\nu) - \alpha(v - v_\nu) = 0,$$

where $(u_\nu, v_\nu) = (x^*_\nu/p^*_\nu, y^*_\nu/p^*_\nu)$ is the image under the inversion transformation of the center c_ν of S_ν, $\nu \in \{j, k\}$. To evaluate the orientation of the center (u_q, v_q) of Z_q, we need to compute the signs of the quantities:

$$o_\nu = \beta(u_q - u_\nu) - \alpha(v_q - v_\nu),$$

which in the inverted coordinates gives:

$$\begin{aligned}
o_\nu\,[(D^u_{jk})^2 + (D^v_{jk})^2] &= (D^v_{jk} D^\rho_{jk} - D^u_{jk}\sqrt{\Gamma})\, D^u_{q\nu} - (D^u_{jk} D^\rho_{jk} + D^v_{jk}\sqrt{\Gamma})\, D^v_{q\nu}\\
&= (D^v_{jk} D^u_{q\nu} - D^u_{jk} D^v_{q\nu})\, D^\rho_{jk} - (D^u_{jk} D^u_{q\nu} + D^v_{jk} D^v_{q\nu})\sqrt{\Gamma}.
\end{aligned} \tag{6}$$

Since $\nu \in \{j, k\}$, it is straightforward to verify that:

$$D^v_{jk} D^u_{q\nu} - D^u_{jk} D^v_{q\nu} = D^{uv}_{qjk} = D^{uv}_{jkq}.$$

Substituting in terms of the original coordinates we have:

$$\begin{aligned}
\Gamma &= (p_j^* p_k^*)^{-2}\,\Delta \\
D_{jk}^{\rho} &= (p_j^* p_k^*)^{-1}\,E_{jk}^{w} \\
D_{jkq}^{uv} &= (p_j^* p_k^* p_q^*)^{-1}\,E_{jkq}^{xy} \\
D_{jk}^{u} D_{q\nu}^{u} + D_{jk}^{v} D_{q\nu}^{v} &= (p_j^* p_k^* p_q^* p_\nu^*)^{-1}\,(E_{jk}^{x} E_{q\nu}^{x} + E_{jk}^{y} E_{q\nu}^{y}) \\
(D_u)^2 + (D_v)^2 &= (p_j^* p_k^*)^{-2}\,[(E_{jk}^{x})^2 + (E_{jk}^{y})^2].
\end{aligned}$$

We can thus rewrite (6) as follows, is terms of the original quantities:

$$o_\nu\,[(E_{jk}^{x})^2 + (E_{jk}^{y})^2] = p_\nu^* E_{jkq}^{xy} E_{jk}^{w} + (E_{jk}^{x} E_{q\nu}^{x} + E_{jk}^{y} E_{q\nu}^{y})\sqrt{\Delta}.$$

To determine the sign of o_ν, we must determine the sign of

$$o'_\nu = p_\nu^* E_{jkq}^{xy} E_{jk}^{w} + (E_{jk}^{x} E_{q\nu}^{x} + E_{jk}^{y} E_{q\nu}^{y})\sqrt{\Delta},$$

which is a quantity of the form $X_0 + X_1\sqrt{Y}$, where the algebraic degrees of X_0, X_1 and Y are 9, 6 and 6, respectively. In fact, X_0 is already factorized into p_ν^*, E_{jkq}^{xy} and E_{jk}^{w}, the algebraic degrees of which are 2, 4, and 3, respectively. Hence, determining the signs of X_0 and X_1 reduces to computing the signs of algebraic expressions of degree at most 6. To deduce the sign of o'_ν, however, we might need to compute the sign of $Z = X_0^2 - X_1^2 Y$, which, a priori, is of degree 18. Below, we will show that Z can be factorized appropriately, thus reducing the algebraic degree of the quantities we need to evaluate in order to determine its sign. Indeed,

$$\begin{aligned}
Z &= (p_\nu^* E_{jkq}^{xy} E_{jk}^{w})^2 - (E_{jk}^{x} E_{q\nu}^{x} + E_{jk}^{y} E_{q\nu}^{y})^2 \Delta \\
&= (p_\nu^* E_{jkq}^{xy})^2 (E_{jk}^{w})^2 - (E_{jk}^{x} E_{q\nu}^{x} + E_{jk}^{y} E_{q\nu}^{y})^2\,[(E_{jk}^{x})^2 + (E_{jk}^{y})^2] + (E_{jk}^{x} E_{q\nu}^{x} + E_{jk}^{y} E_{q\nu}^{y})^2 (E_{jk}^{w})^2 \\
&= [(p_\nu^* E_{jkq}^{xy})^2 + (E_{jk}^{x} E_{q\nu}^{x} + E_{jk}^{y} E_{q\nu}^{y})^2]\,(E_{jk}^{w})^2 - (E_{jk}^{x} E_{q\nu}^{x} + E_{jk}^{y} E_{q\nu}^{y})^2\,[(E_{jk}^{x})^2 + (E_{jk}^{y})^2].
\end{aligned}$$

But,

$$\begin{aligned}
(p_\nu^* E_{jkq}^{xy})^2 + (E_{jk}^{x} E_{q\nu}^{x} + E_{jk}^{y} E_{q\nu}^{y})^2 &= (E_{jk}^{x} E_{q\nu}^{y} - E_{jk}^{y} E_{q\nu}^{x})^2 + (E_{jk}^{x} E_{q\nu}^{x} + E_{jk}^{y} E_{q\nu}^{y})^2 \\
&= (E_{jk}^{x})^2 (E_{q\nu}^{y})^2 + (E_{jk}^{y})^2 (E_{q\nu}^{x})^2 \\
&\quad + (E_{jk}^{x})^2 (E_{q\nu}^{x})^2 + (E_{jk}^{y})^2 (E_{q\nu}^{y})^2 \\
&= [(E_{jk}^{x})^2 + (E_{jk}^{y})^2]\,[(E_{q\nu}^{x})^2 + (E_{q\nu}^{y})^2].
\end{aligned}$$

Hence, we have:

$$\begin{aligned}
Z &= [(E_{jk}^{x})^2 + (E_{jk}^{y})^2]\,[(E_{q\nu}^{x})^2 + (E_{q\nu}^{y})^2]\,(E_{jk}^{w})^2 - (E_{jk}^{x} E_{q\nu}^{x} + E_{jk}^{y} E_{q\nu}^{y})^2\,[(E_{jk}^{x})^2 + (E_{jk}^{y})^2] \\
&= [(E_{jk}^{x})^2 + (E_{jk}^{y})^2]\,\{[(E_{q\nu}^{x})^2 + (E_{q\nu}^{y})^2]\,(E_{jk}^{w})^2 - (E_{jk}^{x} E_{q\nu}^{x} + E_{jk}^{y} E_{q\nu}^{y})^2\}
\end{aligned}$$

Notice that it cannot be the case that $E_{jk}^{x} = E_{jk}^{y} = 0$ (since otherwise X_0 would have been zero and we would have been able to compute the sign of o'_ν without resorting to Z), the sign of Z is the sign of the quantity $[(E_{q\nu}^{x})^2 + (E_{q\nu}^{y})^2]\,(E_{jk}^{w})^2 - (E_{jk}^{x} E_{q\nu}^{x} + E_{jk}^{y} E_{q\nu}^{y})^2$, which is of algebraic degree 12.

A.3 The algebraic degree of the Orientation predicate involving an Apollonius vertex

Suppose we are given three sites $S_\nu = (x_\nu, y_\nu, w_\nu)$, $\nu \in \{i, j, k\}$ and two points $S_\nu = (x_\nu, y_\nu, 0)$, $\nu \in \{l, m\}$, we are interested in computing the orientation $\mathsf{Orientation}(v_{ijk}, S_l, S_m)$, where v_{ijk} is the Apollonius vertex of S_i, S_j and S_k. Emiris and Karavelas [9] have shown that this predicate can be resolved by computing the sign of the quantity

$$\begin{aligned} O :=& 2(E^x_{jk}E^{xw}_{jk} + E^y_{jk}E^{yw}_{jk})E^{xy}_{lm} + (E^y_{jk}D^x_{lm} - E^x_{jk}D^y_{lm})E^w_{jk} \\ &+ (2E^{xy}_{jk}E^{xy}_{lm} - E^x_{jk}D^x_{lm} - E^y_{jk}D^y_{lm})\sqrt{\Delta}, \\ =& (2E^{xy}_{lm}E^{xw}_{jk} - E^w_{jk}D^y_{lm})E^x_{jk} + (2E^{xy}_{lm}E^{yw}_{jk} + E^w_{jk}D^x_{lm})E^y_{jk} \\ &+ (2E^{xy}_{jk}E^{xy}_{lm} - E^x_{jk}D^x_{lm} - E^y_{jk}D^y_{lm})\sqrt{\Delta}. \end{aligned}$$

As for the VertexConflict predicate, it is of the form $X_0 + X_1\sqrt{Y}$, and in order to evaluate its sign we might need to compute the quantity $Z = X_0^2 - X_1^2Y$. As discussed in [9], since the algebraic degrees of X_0, X_1 and Y are 7, 4 and 6 respectively, the Orientation predicate is of algebraic degree 14.

However, as for the VertexConflict predicate we can factorize Z:

$$\begin{aligned} Z =& [(2E^{xy}_{lm}E^{xw}_{jk} - E^w_{jk}D^y_{lm})E^x_{jk} + (2E^{xy}_{lm}E^{yw}_{jk} + E^w_{jk}D^x_{lm})E^y_{jk}]^2 \\ &- (2E^{xy}_{jk}E^{xy}_{lm} - E^x_{jk}D^x_{lm} - E^y_{jk}D^y_{lm})^2[(E^x_{jk})^2 + (E^y_{jk})^2 - (E^w_{jk})^2] \\ =& (2E^{xy}_{lm}E^{xw}_{jk} - E^w_{jk}D^y_{lm})^2(E^x_{jk})^2 + (2E^{xy}_{lm}E^{yw}_{jk} + E^w_{jk}D^x_{lm})^2(E^y_{jk})^2 \\ &+ 2(2E^{xy}_{lm}E^{xw}_{jk} - E^w_{jk}D^y_{lm})(2E^{xy}_{lm}E^{yw}_{jk} + E^w_{jk}D^x_{lm})E^x_{jk}E^y_{jk} \\ &- (2E^{xy}_{jk}E^{xy}_{lm} - E^x_{jk}D^x_{lm} - E^y_{jk}D^y_{lm})^2[(E^x_{jk})^2 + (E^y_{jk})^2] \\ &+ (2E^{xy}_{jk}E^{xy}_{lm} - E^x_{jk}D^x_{lm} - E^y_{jk}D^y_{lm})^2(E^w_{jk})^2 \\ =& (2E^{xy}_{lm}E^{xw}_{jk} - E^w_{jk}D^y_{lm})^2[(E^x_{jk})^2 + (E^y_{jk})^2] + (2E^{xy}_{lm}E^{yw}_{jk} + E^w_{jk}D^x_{lm})^2[(E^x_{jk})^2 + (E^y_{jk})^2] \\ &- (2E^{xy}_{jk}E^{xy}_{lm} - E^x_{jk}D^x_{lm} - E^y_{jk}D^y_{lm})^2[(E^x_{jk})^2 + (E^y_{jk})^2] \\ &- (2E^{xy}_{lm}E^{xw}_{jk} - E^w_{jk}D^y_{lm})^2(E^y_{jk})^2 - (2E^{xy}_{lm}E^{yw}_{jk} + E^w_{jk}D^x_{lm})^2(E^x_{jk})^2 \\ &+ 2(2E^{xy}_{lm}E^{xw}_{jk} - E^w_{jk}D^y_{lm})(2E^{xy}_{lm}E^{yw}_{jk} + E^w_{jk}D^x_{lm})E^x_{jk}E^y_{jk} \\ &+ [(2E^{xy}_{jk}E^{xy}_{lm} - E^x_{jk}D^x_{lm} - E^y_{jk}D^y_{lm})E^w_{jk}]^2 \\ =& [(E^x_{jk})^2 + (E^y_{jk})^2]\, O' + [(2E^{xy}_{jk}E^{xy}_{lm} - E^x_{jk}D^x_{lm} - E^y_{jk}D^y_{lm})E^w_{jk}]^2 \\ &- [(2E^{xy}_{lm}E^{xw}_{jk} - E^w_{jk}D^y_{lm})E^y_{jk} - (2E^{xy}_{lm}E^{yw}_{jk} + E^w_{jk}D^x_{lm})E^x_{jk}]^2 \\ =& [(E^x_{jk})^2 + (E^y_{jk})^2]\, O' + [(2E^{xy}_{jk}E^{xy}_{lm} - E^x_{jk}D^x_{lm} - E^y_{jk}D^y_{lm})E^w_{jk}]^2 \\ &- [2E^{xy}_{lm}(E^{xw}_{jk}E^y_{jk} - E^{yw}_{jk}E^x_{jk}) - (D^x_{lm}E^x_{jk} + D^y_{lm}E^y_{jk})E^w_{jk}]^2, \end{aligned}$$

where

$$O' = (2E^{xy}_{lm}E^{xw}_{jk} - E^w_{jk}D^y_{lm})^2 + (2E^{xy}_{lm}E^{yw}_{jk} + E^w_{jk}D^x_{lm})^2 - (2E^{xy}_{lm}E^{xy}_{jk} - E^x_{jk}D^x_{lm} - E^y_{jk}D^y_{lm})^2.$$

But

$$E^{xw}_{jk}E^y_{jk} - E^{yw}_{jk}E^x_{jk} = E^{xy}_{jk}E^w_{jk},$$

which implies that the last two terms in the last expression for Z above cancel out. Hence, $Z = [(E^x_{jk})^2 + (E^y_{jk})^2]\,O'$. Clearly, the algebraic degree of O' is 8. Moreover, the quantity $(E^x_{jk})^2 + (E^y_{jk})^2$ is strictly positive when we compute the sign of Z, since otherwise X_0 would have been zero (X_0 is a linear combination of $E^x_{jk}$and E^y_{jk}), a case which has already been ruled out according to the procedure in (3). Hence the algebraic degree of the $\mathsf{Orientation}(v_{ijk}, S_l, S_m)$ predicate is 8.

RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399