Qualitative Symbolic Perturbation*

Olivier Devillers[1], Menelaos Karavelas[†2], and Monique Teillaud[3]

1 Inria, Centre de recherche Nancy – Grand Est, France; and
CNRS, Loria, France; and
Université de Lorraine, France

2 Mathematics and Applied Mathematics Department, University of Crete, Greece; and
Institute of Applied and Computational Mathematics, FORTH, Greece

1 Inria, Centre de recherche Nancy – Grand Est, France; and
CNRS, Loria, France; and
Université de Lorraine, France

Abstract

In a classical *Symbolic Perturbation* scheme, degeneracies are handled by substituting some polynomials in ε for the inputs of a predicate. Instead of a single perturbation, we propose to use a sequence of (simpler) perturbations. Moreover, we look at their effects geometrically instead of algebraically; this allows us to tackle cases that were not tractable with the classical algebraic approach.

1998 ACM Subject Classification I.3.5 Computational Geometry and Object Modeling

Keywords and phrases Robustness issues, Symbolic perturbations, Apollonius diagram

Digital Object Identifier 10.4230/LIPIcs.SoCG.2016.33

1 Introduction

In earlier computational geometry papers, the treatment of degenerate configurations was mainly ignored. However, degenerate situations actually do occur in practice. When data are highly degenerate by nature, a direct handling of special cases in a particular algorithm can be efficient [3]. But in many situations, degeneracies happen only occasionally, and perturbation schemes are an easy and efficient generic solution. Controlled perturbations [11] combine increasing arithmetic precision together with actual displacement of the data, and eventually compute a non-degenerate configuration. On the other hand, the use of a symbolic perturbation allows a geometric algorithm or data structure that was originally designed without addressing degeneracies, to still operate on degenerate cases, without concretely modifying the input [7, 14, 15]. Actually, similar strategies were often used by earlier implementors of simple geometric algorithms, without identifying them as symbolic perturbations: for instance when incrementally computing a convex hull, when the new inserted point was lying on a facet of the convex hull, the point was decided to be inside the convex hull.

Let $G(\boldsymbol{u})$ be a geometric structure defined when the input data $\boldsymbol{u}$ satisfies some non-degeneracy assumptions, and let $\boldsymbol{u}_0$ be some input that is degenerate for G. A *symbolic*

[*] 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 Framework (NSRF) – Research Funding Program: THALIS – UOA (MIS 375891).

32nd International Symposium on Computational Geometry (SoCG 2016).
Editors: Sándor Fekete and Anna Lubiw; Article No. 33; pp. 33:1–33:17
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

perturbation consists in using as input $\boldsymbol{u}$ for G a continuous function $\pi(\boldsymbol{u}_0, \varepsilon)$ of a parameter ε. This is done in such a way that for $\varepsilon = 0$, $\pi(\boldsymbol{u}_0, 0)$ is equal to $\boldsymbol{u}_0$, and $\pi(\boldsymbol{u}_0, \varepsilon)$ is non-degenerate for G for sufficiently small positive values of ε. In that case the structure $G(\boldsymbol{u}_0)$ is defined as the limit of $G(\pi(\boldsymbol{u}_0, \varepsilon))$ when $\varepsilon \to 0^+$.

A symbolic perturbation allows an algorithm that computes $G(\boldsymbol{u})$ in generic situations to compute $G(\boldsymbol{u}_0)$ for the degenerate input $\boldsymbol{u}_0$. Most decisions made by the algorithm are usually made by looking at *geometric predicates*, which are combinations of elementary predicates. An elementary predicate is the sign of a continuous real function of the input. The general position assumption is that such a function p never returns 0. When applying a symbolic perturbation, a predicate $\operatorname{sign}(p(\boldsymbol{u}))$ evaluated at $\boldsymbol{u}_0$ returns the limit of $\operatorname{sign}(p(\pi(\boldsymbol{u}_0, \varepsilon)))$ as $\varepsilon \to 0^+$. The sign of $p(\boldsymbol{u}_0)$ can thus be evaluated, provided that $p(\pi(\boldsymbol{u}_0, \varepsilon))$ is not identically equal to 0 for ε in an open interval $(0, a^\star)$ for some $a^\star > 0$. A perturbation scheme is said to be *effective* for a predicate $\operatorname{sign}(p(\boldsymbol{u}))$ if for any $\boldsymbol{u}_0$ the function $\varepsilon \mapsto p(\pi(\boldsymbol{u}_0, \varepsilon))$ is never the null function on any open interval $(0, a)$, with $a \leq a^\star$.

The main difficulty when designing a perturbation scheme for $G(\boldsymbol{u})$ is to find a function $\pi(\boldsymbol{u}_0, \varepsilon)$, such that the perturbation scheme can be proved to be effective for all relevant functions $p(\boldsymbol{u})$, and the perturbed predicates are easy to evaluate, e.g., using as few as possible arithmetic operations. The work of designing and proving the effectiveness of a perturbation for G is typically tailored to a specific algorithm for computing the geometric structure G.

In previous works [1, 6, 7, 8, 12], a predicate is the sign of a polynomial P in some input $\boldsymbol{u} \in \mathbb{R}^m$. The input $\boldsymbol{u}$ is perturbed as an element $\pi(\boldsymbol{u}_0, \varepsilon)$ of $\mathbb{R}^m$ whose coordinates are polynomials in $\boldsymbol{u}_0$ and ε, such that $\pi(\boldsymbol{u}_0, \varepsilon)$ goes to $\boldsymbol{u}_0$ when $\varepsilon \to 0^+$. In the perturbed setting, the predicate returns the sign of the limit $\lim_{\varepsilon \to 0^+} \operatorname{sign}(P(\pi(\boldsymbol{u}_0, \varepsilon)))$. Since P is a polynomial, $P(\pi(\boldsymbol{u}_0, \varepsilon))$ can be rewritten as a polynomial in ε whose monomials in ε are ordered in terms of increasing degree. The constant monomial is actually $P(\boldsymbol{u}_0)$, while the signs of the remaining coefficients can be viewed as auxiliary predicates on $\boldsymbol{u}_0$. The coefficients of $P(\pi(\boldsymbol{u}_0, \varepsilon))$ 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 $\operatorname{sign}(P(\boldsymbol{u}_0))$.

Contribution

In this paper we propose QSP (Qualitative Symbolic Perturbation), a new framework for resolving degenerate configurations in geometric computing. Unlike classical symbolic perturbation techniques, QSP resolves degeneracies in a purely geometric manner, and independently of a specific algebraic formulation of the predicate. So, the technique is particularly suitable for predicates whose algebraic description is not unique or too complicated, such as the ones treated in this paper. In fact, QSP can even handle predicates that are signs of non-polynomial functions.

In addition, instead of having a single perturbation parameter that governs the way the input objects and/or predicates are modified, QSP allows for a sequence of perturbation parameters: 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. To achieve termination, we must 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. The number of objects that need to be perturbed depends on the specific predicate that we analyze. For example in the 2D Apollonius diagram, for a given predicate, perturbing a single object always suffices, whereas in its 3D counterpart, we may need to perturb two input objects.

Standard algebraic symbolic perturbation schemes [7, 8, 12] automatically provide us 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., by factorizing the predicate) is far from being an obvious task. QSP schemes cannot guarantee that the auxiliary predicates are not more complicated algebraically (i.e., are of lower algebraic degree) from the original predicate; however, in principle, the auxiliary predicates that we have to deal with are expected to be more tractable, since their analysis is based on geometric considerations.

As for any perturbation scheme, QSP assumes exact arithmetic to detect degeneracies. Degeneracies are rare enough to allow high efficiency using the exact geometric computing paradigm [16].

In the next section of the paper we formally define the QSP framework. In Section 3 we describe QSP schemes for the main predicates of the 2D Apollonius diagram. In the full version of the paper [5], we apply our technique to the 3D Apollonius diagram and to the arrangement of circles. We end with Section 4, where we discuss the advantages and disadvantages of our framework, and indicate directions for future research.

2 General framework

Let us start with two easy observations about the limit of the sign of a function of two variables.

2.1 Preliminary observations

The first observation allows us to swap the order of evaluation of limits:

▶ **Observation 1.** *Let f be a continuous function of two variables (a, b) defined in a neighborhood of the origin. Let $\lim_{b\to 0^+} \operatorname{sign} f(0, b)$ be denoted as s. If $s \neq 0$ then*

$$\lim_{b\to 0^+} \lim_{a\to 0^+} \operatorname{sign} f(a, b) = s.$$

Proof. Let us assume that $s \neq 0$. There exists $\delta > 0$ such that $\forall b \in (0, \delta]$, $\operatorname{sign} f(0, b) = s$. For any b fixed in $(0, \delta]$ the function $f(a, b)$ is a continuous function in variable a, thus $\lim_{a\to 0^+} f(a, b) = f(0, b)$ and, since $s \neq 0$, f does not vanish when a is in a neighborhood of 0. We have $\lim_{a\to 0^+} \operatorname{sign} f(a, b) = \operatorname{sign} f(0, b) = s$. For any $b \in (0, \delta]$ the function $\lim_{a\to 0^+} \operatorname{sign} f(a, b)$ is the constant function of value s. So its limit is s when $b \to 0^+$. ◀

The second observation formalizes a situation that is in fact trivial.

▶ **Observation 2.** *Let f be a continuous function in two variables (a, b) defined in a neighborhood of the origin. Assume that $\forall b, b' \geq 0$, $\operatorname{sign} f(a, b) = \operatorname{sign} f(a, b')$. Then*

$$\lim_{b\to 0^+} \lim_{a\to 0^+} \operatorname{sign} f(a, b) = \lim_{a\to 0^+} \operatorname{sign} f(a, 0).$$

Proof. Let $s = \lim_{a\to 0^+} \operatorname{sign} f(a, 0)$. There exists $\delta > 0$ such that $\forall a \in (0, \delta]$, $\operatorname{sign} f(a, 0) = s$. By the hypothesis in the observation we have $\forall a \in (0, \delta]$, $\forall b \geq 0$, $\operatorname{sign} f(a, 0) = \operatorname{sign} f(a, b) = s$. The function has constant sign s on $(0, \delta] \times (0, \infty)$ so the limit is s. ◀

2.2 The QSP scheme

Let $G(\boldsymbol{u})$ be a geometric structure whose computation depends on a predicate $\operatorname{sign}(p(\boldsymbol{u}))$, where p is a continuous real function. In this formal presentation, p appears as a function of the whole input $\boldsymbol{u}$; however, in practice, a predicate depends only on a constant size subset of $\boldsymbol{u}$.

We design the perturbation scheme π as a sequence of successive perturbations π_i, $0 \leq i < N$, with

$$\pi(\boldsymbol{u}, \boldsymbol{\varepsilon}) = \pi_0(\pi_1(\pi_2(\dots \pi_{N-1}(\boldsymbol{u}, \varepsilon_{N-1}) \dots, \varepsilon_2), \varepsilon_1), \varepsilon_0),$$

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 numbered by increasing order of magnitude, i.e., 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 $G(\boldsymbol{u})$ to be the limit:

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

QSP implies an evaluation strategy of this limit, as follows. The perturbed predicate

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

is evaluated by first computing $p(\pi(\boldsymbol{u}, (0, 0, \dots, 0)))) = p(\boldsymbol{u})$, and returning its sign if it is non-zero. If $p(\boldsymbol{u}) = 0$, we look at the function $p(\pi(\boldsymbol{u}, (0, 0, \dots, \varepsilon_{N-1}))) = p(\pi_{N-1}(\boldsymbol{u}, \varepsilon_{N-1}))$; 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

$$\ell_1 = \lim_{\varepsilon_{N-1} \to 0^+} \operatorname{sign}(p(\pi_{N-1}(\boldsymbol{u}, \varepsilon_{N-1}))). \tag{1}$$

If ℓ_1 is non-zero, using Observation 1, it is returned as the value of the predicate $\operatorname{sign}(p(\boldsymbol{u}))$. Otherwise, we have to further perturb our geometric input; we examine the limit

$$\ell_2 = \lim_{\varepsilon_{N-1} \to 0^+} \lim_{\varepsilon_{N-2} \to 0^+} \operatorname{sign}(p(\pi_{N-2}(\pi_{N-1}(\boldsymbol{u}, \varepsilon_{N-1}), \varepsilon_{N-2}))). \tag{2}$$

The expression in Eq.(2) can be simplified in cases that actually often occur in applications: if $p(\pi_{N-1}(\boldsymbol{u}, \varepsilon_{N-1})))$ is zero on $[0, \eta)$, it is often also the case that the sign of $p(\pi_{N-2}(\pi_{N-1}(\boldsymbol{u}, \varepsilon_{N-1}), \varepsilon_{N-2}))$ does not depend on ε_{N-1} in $[0, \eta)$. In such a case, and provided that this function is also non-zero, we can evaluate its sign using Observation 2: by taking $\varepsilon_{N-1} = 0$, Eq.(2) boils down to

$$\ell_2 = \lim_{\varepsilon_{N-2} \to 0^+} \operatorname{sign}(p(\pi_{N-2}(\boldsymbol{u}, \varepsilon_{N-2}))). \tag{3}$$

The process is iterated until a non-zero limit is found. In very degenerate situations, when the $\mu - 1$ first limits evaluate to 0, i.e., $\ell_1 = \ell_2 = \ldots = \ell_{\mu-1} = 0$, we need to evaluate ℓ_μ. Its definition is

$$\ell_\mu = \lim_{\varepsilon_{N-1} \to 0^+} \lim_{\varepsilon_{N-2} \to 0^+} \dots \lim_{\varepsilon_{N-\mu} \to 0^+} \operatorname{sign}(p(\pi(\boldsymbol{u}, (0, 0, \dots, 0, \varepsilon_{N-\mu}, \varepsilon_{N-\mu+1}, \dots, \varepsilon_{N-1})))).$$

Similarly to what we described for ℓ_2 above, it is frequently the case that the sign of $p(\pi(\boldsymbol{u}, (0, 0, \dots, 0, \varepsilon_{N-\mu+1}, \dots, \varepsilon_{N-1})))$ does not depend on $\varepsilon_{N-\mu+1}, \dots, \varepsilon_{N-1}$ in a neighborhood of 0 in $\mathbb{R}^{\mu-1}$; then the simplified evaluation allowed by Observation 2 gives:

$$\ell_\mu = \lim_{\varepsilon_{N-\mu} \to 0^+} \operatorname{sign}(p(\pi(\boldsymbol{u}, (0, 0, \dots, 0, \varepsilon_{N-\mu}, \varepsilon_{N-\mu+1}, \dots, \varepsilon_{N-1})))).$$

To assert that the perturbation scheme π is effective, we need to prove that one of these limits is indeed non-zero.

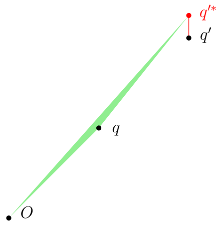
When the predicate is a polynomial, we get a sequence of successive evaluations as in algebraic symbolic perturbations; however, the expressions that need to be evaluated have been obtained in a different way and are *a priori* different. The main advantage of this approach is that we may use a very simple perturbation π_ν, since we do not need each perturbation π_ν to be effective, but rather the composed perturbation π. For geometric problems, the simplicity of π_ν allows us to look at the limit in a geometric manner, instead of algebraically computing some appropriate coefficient of $p(\pi(\boldsymbol{u},\varepsilon))$.

2.3 Toy examples

We illustrate these principles with three toy examples. In all examples, we set $\boldsymbol{u} = (q, q') = ((x_0, x_1), (x_2, x_3))$, a pair of two 2D points and $\pi_i(\boldsymbol{u}, \varepsilon_i) = \boldsymbol{u} + \varepsilon_i \boldsymbol{e}_i$, where $\boldsymbol{e}_0 = ((1, 0), (0, 0))$, $\boldsymbol{e}_1 = ((0, 1), (0, 0))$, $\boldsymbol{e}_2 = ((0, 0), (1, 0))$, and $\boldsymbol{e}_3 = ((0, 0), (0, 1))$ form the canonical basis of

$(\mathbb{R}^2)^2$. The differences between the examples below lie in the evaluated predicate $\text{sign}(p(\boldsymbol{u}))$ and the degenerate position $\boldsymbol{u}_0$.

First example: orientation of a flat triangle



Let $p(\boldsymbol{u}) = x_0x_3 - x_1x_2$ and $\boldsymbol{u}_0 = (q, q') = ((1, 1), (2, 2))$. QSP defines the result for $\text{sign}(p(\boldsymbol{u}_0))$, the orientation of Oqq', as

$$\text{sign}(p(\boldsymbol{u}_0)) \quad = \quad \lim_{\varepsilon_3\to 0^+} \lim_{\varepsilon_2\to 0^+} \lim_{\varepsilon_1\to 0^+} \lim_{\varepsilon_0\to 0^+} \text{sign}((1+\varepsilon_0)(2+\varepsilon_3) - (2+\varepsilon_1)(1+\varepsilon_2)).$$

A standard evaluation of this expression would consist in taking the limits in order:

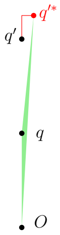
$$\begin{aligned}\text{sign}(p(\boldsymbol{u}_0)) &= \lim_{\varepsilon_3\to 0^+} \lim_{\varepsilon_2\to 0^+} \lim_{\varepsilon_1\to 0^+} \text{sign}((2+\varepsilon_3) - (2+\varepsilon_1)(1+\varepsilon_2)) \\ &= \lim_{\varepsilon_3\to 0^+} \lim_{\varepsilon_2\to 0^+} \text{sign}((2+\varepsilon_3) - 2(1+\varepsilon_2)) = \lim_{\varepsilon_3\to 0^+} \text{sign}(\varepsilon_3) = 1.\end{aligned}$$

Following the QSP evaluation strategy instead, in such a case, the biggest perturbation, i.e., the perturbation on x_3, allows to quickly conclude. The only computed limit is the one in Eq. (1):

$$\ell_1 = \lim_{\varepsilon_3\to 0^+} \text{sign}(p((1, 1), (2, 2+\varepsilon_3))) = \lim_{\varepsilon_3\to 0^+} \text{sign}((2+\varepsilon_3) - 2) = \lim_{\varepsilon_3\to 0^+} \text{sign}(\varepsilon_3) = 1.$$

The geometric interpretation is that we get the orientation of a triangle Oqq'^* for a point q'^* slightly above q'.

Second example: orientation of a vertical flat triangle



Let $p(\boldsymbol{u}) = x_0x_3 - x_1x_2$ and $\boldsymbol{u}_0 = (q, q') = ((0,1),(0,2))$. QSP defines the result for $\mathrm{sign}(p(\boldsymbol{u}_0))$, the orientation of Oqq', as

$$\mathrm{sign}(p(\boldsymbol{u}_0)) = \lim_{\varepsilon_3\to 0^+}\lim_{\varepsilon_2\to 0^+}\lim_{\varepsilon_1\to 0^+}\lim_{\varepsilon_0\to 0^+}\mathrm{sign}((0+\varepsilon_0)(2+\varepsilon_3)-(1+\varepsilon_1)(0+\varepsilon_2)).$$

Taking the limits in order leads to:

$$\begin{aligned}\mathrm{sign}(p(\boldsymbol{u}_0)) &= \lim_{\varepsilon_3\to 0^+}\lim_{\varepsilon_2\to 0^+}\lim_{\varepsilon_1\to 0^+}\mathrm{sign}(-(1+\varepsilon_1)\varepsilon_2)\\ &= \lim_{\varepsilon_3\to 0^+}\lim_{\varepsilon_2\to 0^+}\mathrm{sign}(-\varepsilon_2)) = \lim_{\varepsilon_3\to 0^+}\mathrm{sign}(-1) = -1.\end{aligned}$$

In this case, the QSP evaluation strategy is to first compute

$$\ell_1 = \lim_{\varepsilon_3\to 0^+}\mathrm{sign}(p((0,1),(0,2+\varepsilon_3))) = \lim_{\varepsilon_3\to 0^+} 0 = 0,$$

which does not allow us to resolve the degeneracy. Then we observe that

$$\mathrm{sign}(p((0,1),(x_2,x_3)) = \mathrm{sign}(-x_2)$$

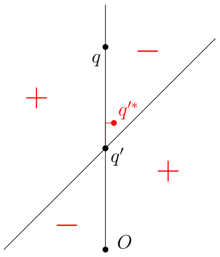
does not depend on x_3, thus we can evaluate ℓ_2 using Eq. (3):

$$\ell_2 = \lim_{\varepsilon_2\to 0^+}\mathrm{sign}(p((0,1),(\varepsilon_2,2))) = \lim_{\varepsilon_2\to 0^+}\mathrm{sign}(-\varepsilon_2) = -1.$$

Two perturbations π_3 and π_2 must be used, but the simplified evaluation of ℓ_2 suffices.

The geometric interpretation is that we look at the orientation of a triangle Oqq'^* for a moved point q'^*. Since moving q'^* slightly above q' doesn't change anything to the degeneracy, the point is moved to the right, which resolves the degeneracy.

Third example: points and quadratic form



Let $p(\boldsymbol{u}) = x_0(x_1-1) - x_0^2 - x_2(x_3-1) + x_2^2$ and $\boldsymbol{u}_0 = (q, q') = ((0,2),(0,1))$. The predicate p stands for the difference of a degenerate quadratic form evaluated at q and q'. QSP defines the result for $\mathrm{sign}(p(\boldsymbol{u}_0))$ as

$$\mathrm{sign}(p(\boldsymbol{u}_0)) = \lim_{\varepsilon_3\to 0^+}\lim_{\varepsilon_2\to 0^+}\lim_{\varepsilon_1\to 0^+}\lim_{\varepsilon_0\to 0^+}\mathrm{sign}\left(\varepsilon_0(1-2+\varepsilon_1)-\varepsilon_0^2-\varepsilon_2(1-1+\varepsilon_3)+\varepsilon_2^2\right),$$

which could be evaluated as follows:

$$\begin{aligned}\operatorname{sign}(p(\boldsymbol{u}_0)) &= \lim_{\varepsilon_3\to 0^+}\lim_{\varepsilon_2\to 0^+}\lim_{\varepsilon_1\to 0^+}\operatorname{sign}\left(-\varepsilon_2\varepsilon_3+\varepsilon_2^2\right)\\ &= \lim_{\varepsilon_3\to 0^+}\lim_{\varepsilon_2\to 0^+}\operatorname{sign}\left(\varepsilon_2(\varepsilon_2-\varepsilon_3)\right) = \lim_{\varepsilon_3\to 0^+}\operatorname{sign}(-\varepsilon_3) = -1.\end{aligned}$$

Again the evaluation strategy first computes

$$\ell_1 = \lim_{\varepsilon_3\to 0^+}\operatorname{sign}(p((0,2),(0,1+\varepsilon_3))) = \lim_{\varepsilon_3\to 0^+} 0 = 0,$$

which does not allow us to resolve the degeneracy.

Then we observe that $\operatorname{sign}(p((0,2),(x_2,x_3))) = \operatorname{sign}(x_2(x_3-1)+x_2^2)$ actually depends on x_3, thus we must evaluate ℓ_2 using Eq. (2):

$$\begin{aligned}\ell_2 &= \lim_{\varepsilon_3\to 0^+}\lim_{\varepsilon_2\to 0^+}\operatorname{sign}(p((0,2),(\varepsilon_2,1+\varepsilon_3)))\\ &= \lim_{\varepsilon_3\to 0^+}\lim_{\varepsilon_2\to 0^+}\operatorname{sign}\left(\varepsilon_2(\varepsilon_2-\varepsilon_3)\right) = \lim_{\varepsilon_2\to 0^+}\operatorname{sign}(-\varepsilon_3) = -1.\end{aligned}$$

Notice that since $\operatorname{sign}(p((2,0),(x_2,x_3)))$ depends on x_3, the simplified evaluation of Eq. (3) would have given a wrong result:

$$\ell_2 \neq \lim_{\varepsilon_2\to 0^+}\operatorname{sign}(p((2,0),(\varepsilon_2,1))) = \lim_{\varepsilon_2\to 0^+}\operatorname{sign}(\varepsilon_2^2) = 1.$$

The geometric interpretation is that q and q' are both on one of the two lines defined by the quadratic equation $x(y-1)-x^2=0$. Point q' is first slightly moved upwards but this motion leaves it on that same line, then it is moved to the right, and the sign of the quadratic form depends on the vertical position of q' with respect to the other line.

A fourth toy example illustrating a non-polynomial predicate is given in the full version [5].

2.4 Discussion

Multiple epsilons

The idea of utilizing multiple perturbation parameters is already present in Yap's scheme [14], or very recently in Irving and Green's work [10], but without the geometric interpretation allowed by QSP. In other previous works, such as SoS [7], the algebraic symbolic perturbation framework was proved to be effective by a careful choice of the exponents for ε, depending on the choice of $G(\boldsymbol{u})$, so as to make some terms negligible. QSP can be forced to fit in such a traditional framework, with a single epsilon, by making all the variables ε_ν dependent on a single parameter κ that plays the traditional role of ε. For polynomial predicates, 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 $0\le\nu<N-1$. The interest of QSP, however, is not to use this traditional view, but rather have the variables ε_ν independent; this decoupling allows for additional flexibility, and, in particular, permits us to think of the sequence of perturbations in geometric terms.

Efficiency

The aim of a perturbation scheme is to solve degeneracies, and a common assumption is that such degeneracies are rare enough so that some extra time can be spent to make a reliable decision when a degeneracy happens. Another implicit assumption is that degeneracies are actually detected, that is, it is implicitly assumed that the original predicates are computed exactly, possibly with some filtering mechanism to ensure efficiency [16].

Nevertheless, the actual additional complexity in case of degeneracy must be addressed. Since QSP is geometrically defined and addresses very general problems, such a complexity analysis cannot be done at the general level. For the two applications described in this paper, the extra predicates needed to resolve the degeneracy have the same complexity as the original ones, while the number of epsilons used to perturb is not bigger than two.

QSP, as many other perturbation schemes, relies on an indexing of the input. However, as mentioned earlier, a given predicate usually depends on a constant number of input objects. It is important to keep in mind that the comparison of indices is necessary only for the few objects involved in a given predicate; sorting the whole input with respect to indices is not required.

Generality

In the first three toy examples above, SoS would have taken $\varepsilon_0 = \varepsilon^8$, $\varepsilon_1 = \varepsilon^4$, $\varepsilon_2 = \varepsilon^2$, and $\varepsilon_3 = \varepsilon$, which yields the same result as QSP. When it leads to a simple result, the classical algebraic view is a very good solution. However, if the original predicate is a bit intricate, the algebraic way will produce numerous extra predicates to resolve degeneracies. Moreover, as for any predicate, some custom work is often still needed on the polynomial to evaluate it efficiently, e.g., finding a good factorization.

QSP provides a very general approach that is able to handle various predicates, even non-polynomial (as in the fourth example described in [5]). Of course applying this scheme to a given problem requires some problem-specific work, but, as noted in the previous paragraph, this is also often the case for the above-mentioned algebraic approaches. In algebraic approaches, obtaining the coefficient of ε^i in a suitable way for an efficient evaluation is a non-trivial task; the task is even harder when the predicate does not boil down to evaluating a single polynomial, as it is the case for Apollonius predicates, which we present in the sequel.

We would not advise the use of QSP for simple cases such as Delaunay triangulations of points where other approaches work well [6], but rather only in cases where the predicates are very complex or non-polynomial. The applications below use high degree polynomials and QSP is a good solution. As far as we know, no other perturbation scheme has ever been proposed for Apollonius diagrams. Regarding intersections of circles, we successfully addressed the predicate comparing the abscissas of intersection of circles, using QSP [5]. The only other result that we know of for perturbing this predicate was recently obtained by Irving and Green [10]; it uses a pseudo-random scheme.

Meaningfulness

According to a classification by Seidel [13], QSP is (geometrically) meaningful, that is we have some control on the direction (in input data space) used to move away from the degeneracies. For example, for the Apollonius diagram we will choose to minimize the number of Apollonius vertices (it is also possible to choose to maximize it). QSP is not independent of indexing, but if this indexing is geometrically meaningful, then we can ensure invariance with respect to some geometric transformations.

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 $\mathbb{R}^d$. In the formalism of Section 2, $\boldsymbol{u}$ is a vector

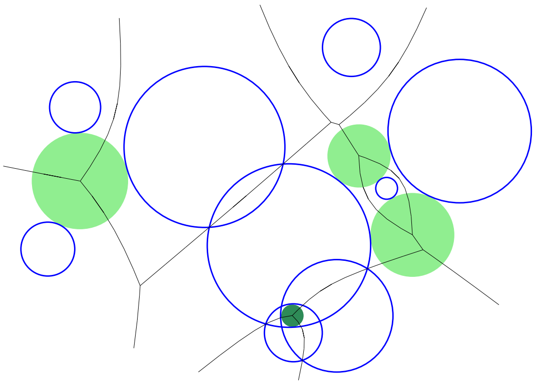


Figure 1 Planar Apollonius diagram. Weighted points are in light blue. A green disk is centered at an Apollonius vertex and its radius is the weighted distance of the center to its three closest sites. The darker green disk has a negative distance to its closest neighbors.

of coordinates and weights of a set of weighted points and $G(\boldsymbol{u})$ is the Apollonius diagram; in this section, we will use notations more adapted to our application. The Euclidean norm is denoted as $|\cdot|$. The *weighted distance* from a query point r to a weighted point (s, w), where s is a point in the Euclidean space and $w \in \mathbb{R}$, is $|rs| - w$. 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 with the sphere S centered at s and of radius w. The distance from a query point r to a site $S = (s, w)$ is the Euclidean distance from r to S, with a negative sign if r lies inside S.

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 sphere S_i, for all $i = 0, \ldots, d$, and *internal* if it is inside the spheres. 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 figure above). It is always possible to ensure an external configuration locally 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.

Let us show that $d+1$ sites in general position define 0, 1 or 2 Apollonius vertices. Assume, without loss of generality, that all weights are non-negative for $i = 1, \ldots, d$ and $w_0 = 0$, so as to have an external configuration. Consider now the inversion with point s_0 as the pole. The point s_0 goes to infinity, while each sphere S_i, $i = 1, \ldots, d$ becomes a new sphere $Z_i = (z_i, \rho_i)$. Determining the balls B_α (where α indexes the different solutions) tangent to the spheres S_i, $i = 0, \ldots, d$ is equivalent to determining halfspaces delimited by the hyperplanes T_α tangent to the spheres Z_i, $i = 1, \ldots, d$, with all spheres on the same side of T_α. Requiring that a given B_α is externally tangent to the spheres S_i is equivalent to requiring that T_α separates the spheres Z_i from the origin. The normalized equation of T_α: $\lambda_\alpha \cdot x + \delta_\alpha = 0$, with $\lambda_\alpha \in \mathbb{R}^d$, $|\lambda_\alpha| = 1$ and $\delta_\alpha \in \mathbb{R}$, gives the signed distance of a point $x \in \mathbb{R}^d$ to T_α. We have

$$T_\alpha \text{ tangent to } Z_i,\ 1 \leq i \leq d \quad \Longleftrightarrow \quad \left\{ \begin{array}{l} \lambda_\alpha \cdot z_i + \delta_\alpha = \rho_i,\ 1 \leq i \leq d \\ |\lambda_\alpha|^2 = 1 \end{array} \right. . \tag{4}$$

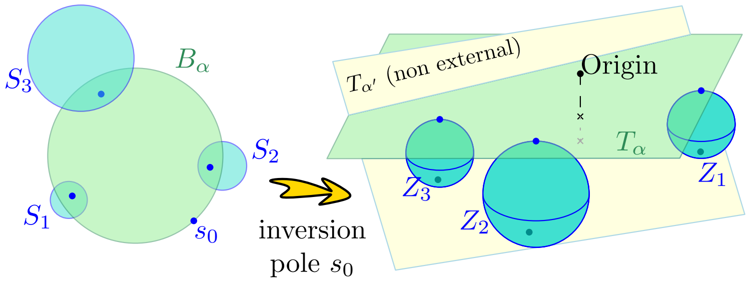


Figure 2 The spheres B_α externally tangent to the sites S_i, $i = 0, \ldots, d$, correspond, via the inversion transformation with s_0 as the pole, to hyperplanes T_α tangent to the spheres Z_i that separate them from the origin.

In the inverted space, the general position hypothesis means that the spheres Z_i do not have an infinity of tangent hyperplanes; the latter can occur only if the points z_i (and thus the points s_i) are affinely dependent. Therefore, the system (4) of one quadratic and d linear equations in $d+1$ unknowns ($\lambda_\alpha \in \mathbb{R}^d$ and $\delta_\alpha \in \mathbb{R}$) has at most two real solutions by Bézout's theorem, hence the first claim follows. Depending on the position of the origin with respect to T_α (or equivalently on the sign of δ_α), zero, one or both solutions may correspond to external configurations.

An Apollonius vertex is actually defined by a sequence of $d+1$ sites in general position, up to a positive permutation of the sequence. Indeed, in the previous paragraph, if there are two solutions T_α and $T_{\alpha'}$, we observe that they are symmetric with respect to the hyperplane spanned by the points z_i, thus the d-simplex formed by the tangency points and the origin has different orientations for the two solutions. This implies that the two solutions can be distinguished by the signature of the permutation of the spheres S_i.

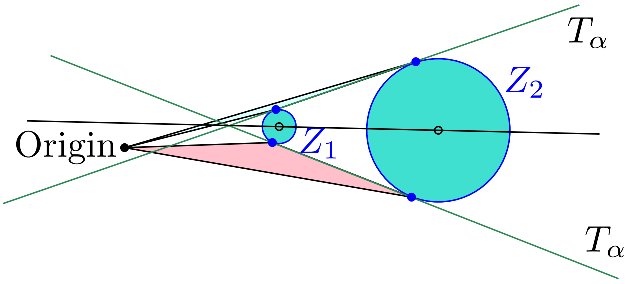


Figure 3 If T_α and $T_{\alpha'}$ are both external, the simplices formed by the tangency points and the origin have different orientations.

3.2 The VConflict predicate

Several predicates are necessary to compute an Apollonius diagram. We start with the *vertex conflict* predicate $\mathsf{VConflict}(\boldsymbol{S}^v, Q)$, which 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 w_q for Q, which was *a priori* excluded above, but is treated below.

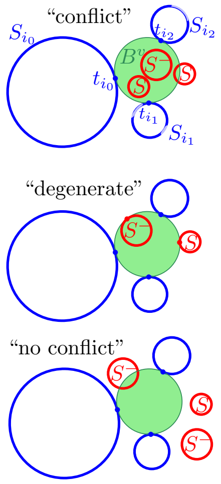
We denote by $B_{i_0 i_1 \ldots i_d}$ the open ball whose closure $\overline{B}_{i_0 i_1 \ldots i_d}$ is tangent to the sites of $\boldsymbol{S}^v$. The contact points $t_{i_0}, t_{i_1}, \ldots, t_{i_d}$ define a positively oriented d-simplex.

If $w_q \geq 0$, the predicate $\mathsf{VConflict}(\boldsymbol{S}^v, Q)$ answers

- “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 $w_q < 0$, we define Q^- as the sphere with the same center s_q as Q and radius $-w_q$. Then $\mathsf{VConflict}(\boldsymbol{S}^v, Q)$ answers

- “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.



Qualitative perturbation of the VConflict predicate

QSP relies on some ordering of the sites. Each site $S_\nu = (s_\nu, w_\nu)$ is perturbed to $S_\nu^\varepsilon = (s_\nu, w_\nu + \varepsilon_\nu)$, $\varepsilon_\nu \geq 0$, with S_λ perturbed more than S_ν if $\lambda > \nu$. Following the QSP framework, if the configuration is still degenerate after we have enlarged the site of maximum index, then we enlarge the site with the second largest index, and so on. As mentioned in the general presentation (Section 2.4), we need only consider the sites involved in the predicate, and enlarge them one-by-one until the resulting configuration is non-degenerate, in which case the predicate is resolved. Sites are sorted internally in the predicate, among a constant number of objects; there is no need for sorting the sites, with respect to their index, globally.

Any indexing can be used. We choose what we call the *max-weight* indexing 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 weights, until the degeneracy is resolved. To break ties between sites with the same weights, we use the lexicographic comparison of their centers: among two sites with the same weight, the site whose center is lexicographically smaller than the other is assigned a smaller max-weight index. The max-weight indexing has the strong advantage of being geometrically meaningful. It favors sites with larger weights, so, if two sites are internally tangent, then the site with the larger weight will be perturbed more, in which case the site with the smallest weight will be inside the interior of the other site, and its Apollonius region will disappear in the perturbed diagram. As a first consequence, this indexing 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 sites 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. Some inevitable degenerate constructions, such as zero-length Apollonius edges, are handled seamlessly by the method. As a final comment, the max-weight scheme can be used to resolve the degeneracies of all predicates described by Emiris and Karavelas for the 2D case [9].

3.3 Perturbing circles for the 2D Apollonius diagram

In two dimensions, the two main predicates for computing Apollonius diagrams are the VConflict predicate introduced in the previous section and the EdgeConflict predicate. Predicate EdgeConflict is presented in the full version of the paper [5].

Given S_i, S_j, S_k the three sites that define an Apollonius circle in the Apollonius diagram and $Q = S_q$ the query site, the algebraic formulation of predicate VConflict is a polynomial in the coordinates and weights of the four spheres. Emiris and Karavelas [9] proposed a degree 14 expression that can be factorized to reduce the degree to 8 [5, Appendix B].

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

3.3.1 Algebraic perturbation of the 2D VConflict predicate

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 the algebraic polynomial involved in VConflict will give a polynomial in ε_i, ε_j, ε_k and ε_q with hundreds of terms (see full version [5] for more details). Assigning ε_i, ε_j, ε_k and ε_q to be polynomial functions of a single variable ε (for example, we may set $\varepsilon_\nu = \varepsilon^{\alpha_\nu}$, $\nu \in \{i, j, k, q\}$) transforms the expression to 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 variable ε_ν in the resulting polynomial do not collapse. Even if one could find an assignment that does not make the coefficients (of the originally different terms) 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 2D VConflict predicate

We now precisely describe how the perturbation works on the VConflict predicate in dimension 2. Let us denote by q the max-weight index of Q, i.e., $Q = S_q$. We denote with superscript ε the perturbed version of objects, that is B_{ijk}^ε is a shorthand for the ball tangent to S_i^ε, S_j^ε, and S_k^ε, and

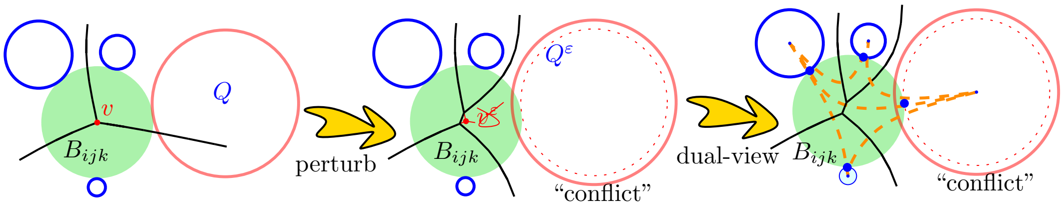
$$\mathsf{VConflict}^\varepsilon(S_i, S_j, S_k, S_q) = \lim_{\varepsilon_{i_3}\to 0^+} \lim_{\varepsilon_{i_2}\to 0^+} \lim_{\varepsilon_{i_1}\to 0^+} \lim_{\varepsilon_{i_0}\to 0^+} \mathsf{VConflict}(S_i^\varepsilon, S_j^\varepsilon, S_k^\varepsilon, Q^\varepsilon)$$

with $i_0 < i_1 < i_2 < i_3$, $\{i_0, i_1, i_2, i_3\} = \{i, j, k, q\}$.

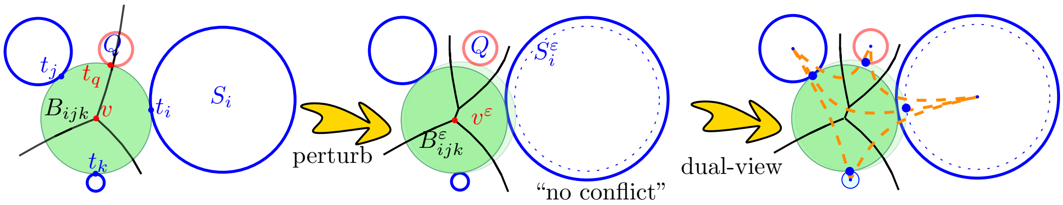
If $q > i, j, k$ and $\mathsf{VConflict}\ (S_i, S_j, S_k, Q) =$ "degenerate", we compute the limit given by Eq. (1)

$$\lim_{\varepsilon_q\to 0^+} \mathsf{VConflict}(S_i, S_j, S_k, Q^\varepsilon).$$

It is clear that this limit always evaluates to "conflict", since Q is growing while the open ball B_{ijk} whose closure is tangent to S_i, S_j, and S_k can be considered as fixed and we do not need to look at perturbations of smaller index.

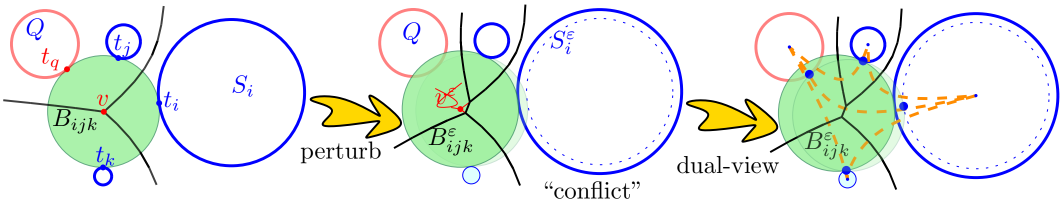


If q is not the largest index, then B_{ijk} can be viewed as defined by three other circles among S_i, S_j, S_k, and Q. Since $B_{ijk} = B_{jki} = B_{kij}$, we can assume, without loss of generality, that $i > j, k, q$. Moreover, B_{ijk} coincides with either B_{jkq} or B_{kjq}, depending on the orientation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



In the perturbed setting, S_i^ε is in conflict with B_{jkq}^ε (or B_{kjq}^ε) since S_i^ε is growing, while B_{jkq}^ε can be considered as fixed. We simply need to determine if B_{ijk}^ε remains empty in the perturbed setting. 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,[1] 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 triangle $S_i S_j S_k$ and, therefore, Q is not in conflict with B_{ijk} (see figure above), otherwise $S_i S_j S_k$ is not in the triangulation and Q has to be in conflict with B_{ijk} (see figure below).

[1] The dual of the Apollonius diagram is called Apollonius graph. The Apollonius region of S_i is associated to a vertex of the dual graph, thus S_i can be used to refer to the corresponding vertex in the dual graph.



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^ε_{ijk} and thus the result of the perturbed predicate is "no conflict"; in the latter case, Q intersects B^ε_{ijk}, 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 2D $\mathsf{VConflict}^\varepsilon$ predicate

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

```
1. if $\mathsf{VConflict}(S_i, S_j, S_k, Q) \neq$ "degenerate" then return $\mathsf{VConflict}(S_i, S_j, S_k, Q)$;
2. if $q > \max\{i, j, k\}$ then return "conflict";
3. ensure that $i > \max\{j, k\}$ 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 $q > j$ then return "conflict"; else return "no conflict"; };
6. if $t_q = t_k$ then { if $q > k$ then return "conflict"; else return "no conflict"; };
7. if $t_j t_k t_q$ is ccw then return "no conflict"; else return "conflict";
```

Step 1 is the unperturbed predicate and is evaluated algebraically, e.g., as described in [9, 5]. 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 a similar 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.

We will now focus on this last step, Step 7, because it introduces a new geometric predicate, which is difficult to evaluate: $\mathsf{Orientation}(t_j, t_k, t_q)$, for three tangency points. Our aim is to reduce the complexity of the expressions to be evaluated, which is why we avoid computing the tangency points explicitly. The end of this section describes a method with algebraic degree 8, as in Step 1. This computation can be done in another way: in [5, Appendix A] we proposed an alternative method that requires very few arithmetic computations besides the quantities already computed in the unperturbed evaluation of Step 1, however these few computations have algebraic degree 12.

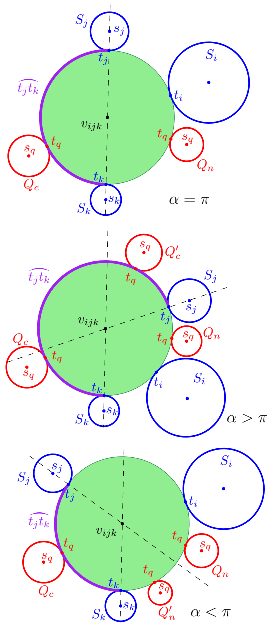
It has been shown in [9] that evaluating the orientation of three points where two are centers of sites and the third is an Apollonius vertex, can be performed using algebraic expressions of degree at most 14. In fact, this degree may be decreased to 8 (see [5, Appendix B]), in which case we resolve $\mathsf{Orientation}(t_j, t_k, t_q)$, without resorting to a higher degree predicate, as described below.

Firstly, 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 than π. Secondly, we distinguish between the following cases:

$\boldsymbol{o_1}$ **= "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: $\mathsf{Orientation}(t_j, t_k, t_q) = \mathsf{Orientation}(s_j, s_k, s_q)$ (see Q_n (resp., Q_c) in the figure (top) to illustrate a position of Q not in conflict (resp., in conflict)).

$\boldsymbol{o_1}$ **= "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 the figure (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 the figure (middle)).

$\boldsymbol{o_1}$ **= "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 the figure (bottom)). 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 the figure (bottom)).



To summarize, the evaluation of Step 7 requires at most three orientation tests involving an Apollonius vertex and two sites; one may be obtained as a subproduct of Step 1, while the other two require work similar to the work performed for Step 1. Thus, the evaluation of Step 7 does not increase the algebraic degree of the $\mathsf{VConflict}$ predicate.

4 Conclusion

In this extended abstract, a new framework for dealing with geometric degeneracies has been proposed: QSP, and its application to a predicate used in the computation of the 2D Apollonius diagram. In the full paper we extend the result to the 3D Apollonius diagram and to the computation of arrangement of arcs of circles in the plane. All these predicates are predicates of medium degree (8 to 28) and have complicated algebraic expressions that make them difficult to combine with classical algebraic perturbation schemes. 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 QSP 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 design for the perturbed predicate (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 decreasing weight, whereas for the trapezoidal map of circular arcs we first consider a (global) rotation and then the circles by means of decreasing 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. Another type of geometric problem, involving complex predicates, for which the QSP framework is relevant, is the computation of lines tangent to four given lines in 3D [2, 4].

References

1 P. Alliez, O. Devillers, and J. Snoeyink. Removing degeneracies by perturbing the problem or the world. *Reliable Computing*, 6:61–79, 2000. URL: http://hal.inria.fr/inria-00338566/.

2 H. Brönnimann, O. Devillers, V. 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. URL: http://hal.inria.fr/inria-00103916.

3 C. Burnikel, K. Mehlhorn, and S. Schirra. On degeneracy in geometric computations. In *5th ACM-SIAM Sympos. Discrete Algorithms*, pages 16–23, 1994. URL: http://dl.acm.org/citation.cfm?id=314474.

4 O. Devillers, M. Glisse, and S. 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. URL: http://hal.inria.fr/inria-00336256/.

5 O. Devillers, M. Karavelas, and M. Teillaud. Qualitative symbolic perturbation: two applications of a new geometry-based perturbation framework. Research Report 8153, INRIA, 2015. version 4. URL: http://hal.inria.fr/hal-00758631/.

6 O. Devillers and M. Teillaud. Perturbations for Delaunay and weighted Delaunay 3D triangulations. *Computational Geometry: Theory and Applications*, 44:160–168, 2011. doi: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. URL: http://dl.acm.org/citation.cfm?id=77639.

8 I. Emiris and J. Canny. A general approach to removing degeneracies. *SIAM J. Comput.*, 24:650–664, 1995. URL: http://epubs.siam.org/sicomp/resource/1/smjcat/v24/i3/p650_s1.

9 I. Emiris and M. Karavelas. The predicates of the Apollonius diagram: algorithmic analysis and implementation. *Computational Geometry: Theory and Applications*, 33(1-2):18–57, January 2006. doi:10.1016/j.comgeo.2004.02.006.

10 G. Irving and F. Green. A deterministic pseudorandom perturbation scheme for arbitrary polynomial predicates. Technical Report 1308.1986v1, arXiv, 2013. URL: http://arxiv.org/abs/1308.1986.

11 K. Mehlhorn, R. Osbild, and M. Sagraloff. A general approach to the analysis of controlled perturbation algorithms. *Comput. Geom. Theory Appl.*, 44:507–528, 2011. doi:10.1016/j.comgeo.2011.06.001.

12 R. Seidel. The nature and meaning of perturbations in geometric computing. *Discrete Comput. Geom.*, 19:1–17, 1998.

13 R. Seidel. Perturbations in geometric computing, 2013. Workshop on Geometric Computing, Heraklion. URL: http://www.acmac.uoc.gr/GC2013/files/Seidel-slides.pdf.

14 C. K. Yap. A geometric consistency theorem for a symbolic perturbation scheme. *J. Comput. Syst. Sci.*, 40(1):2–18, 1990. URL: http://www.sciencedirect.com/science/article/pii/002200009090016E.

15 C. K. Yap. Symbolic treatment of geometric degeneracies. *J. Symbolic Comput.*, 10:349–370, 1990. URL: http://www.sciencedirect.com/science/article/pii/S0747717108800697.

16 C. K. Yap and T. Dubé. The exact computation paradigm. In *Computing in Euclidean Geometry*, volume 4 of *Lecture Notes Series on Computing*, pages 452–492. World Scientific, 1995. URL: http://www.cs.nyu.edu/~exact/doc/paradigm.ps.gz.