Removing degeneracies by perturbing the problem or perturbing the world

Pierre Alliez, Olivier Devillers, Jack Snoeyink

Removing degeneracies by perturbing the problem or perturbing the world*

Pierre Alliez
CCETT/DIH/ATI, Rennes
pierre.alliez@cnet.francetelecom.fr

Olivier Devillers
INRIA Sophia Antipolis
BP 93, 06902 Sophia Antipolis, France
Olivier.Devillers@sophia.inria.fr

Jack Snoeyink
Computer Science, UBC, Vancouver Canada and
Computer Science, UNC Chapel Hill
Campus Box 3175, Sitterson Hall
Chapel Hill, N.C. 27599–3175 USA
snoeyink@cs.unc.edu

July 7, 1999

Abstract

We describe two problem-specific approaches to remove geometric degeneracies that we call *perturbing the problem* and *perturbing the world*. Using as our primary examples 2-d and 3-d Delaunay triangulation with Euclidean and polygonal metrics, we show that these approaches lead to relatively simple and efficient perturbations of the points that do not depend on a fixed ordering or index. Thus, they produce canonical output, which is important for producing test suites and verifiers for randomized or dynamic geometric algorithms.

1 Introduction

Algorithms in computational geometry typically infer topological structures from geometric tests that are performed on numerical coordinates. When designing algorithms it is common to assume

*This work was partially supported by ESPRIT IV LTR Project No 28155 (Galia). The third author gratefully acknowledges support from NSERC and, while visiting INRIA on leave, also from CIES and the Killam Foundation.

that the geometric entities are in *general position*—that small subsets do not satisfy any of a chosen finite set of algebraic conditions that would, with probability one, no longer be satisfied if the entities were randomly perturbed. Example assumptions in the plane include that no two points lie on the same vertical line, that no three points are collinear, or that no four points are co-circular.

Subsets that violate these assumptions are called *degeneracies* or *degenerate cases*. Which things are considered degeneracies depend on the problem or algorithm used for its solution; sweep algorithms may need to detect and handle points on the same vertical line, while other algorithms may not care.

Degeneracies can be resolved by locally perturbing the input so that the assumption holds. However, it can be dangerous to make arbitrary local decisions. Consider the problem of sorting with equal keys. If one decides that equal keys compare as "less than" then standard implementations of Quicksort, such as the Unix `qsort`(), will go into an infinite loop. Simple exchange or insertion sort algorithms will not exhibit this problem, but an efficient algorithm like Quicksort depends on a comparison that is consistent with an order on the elements: it assumes that if $a < b$, then $b \not< a$.

Bugs that occur due to improper detection or handling of degenerate cases can be difficult to reproduce and, therefore, track down and fix. Furthermore, it is tedious and time-consuming to design test suites to cover all the degenerate cases. Thus, as computational geometry turns more to implementation of its algorithms, it is not surprising that there is much interest in methods to automatically handle degeneracies.

Perturbation techniques, which replace a set P of n input points by another set Q that has no degeneracies, but is 'close' (often' infinitesimally close') to P. More formally, there will be a $\delta > 0$ and sets Q_ϵ, for each $0 < \epsilon < \delta$, such that Q_ϵ has no degenerate configurations and each point of P has exactly one point of Q_ϵ within distance ϵ.

There are a number of general perturbation schemes, which we review in the next section. In this paper, however, we describe an approach to design and implement problem-specific perturbation schemes. Our approach has three steps: First, perturb the problem—rather than solving the original problem in a degenerate case, consider a closely-related problem. Second, perturb the world—extract from the perturbed problem a perturbation of the input, and prove that it resolves all degenerate cases. Third, perturb the code—implement tests that handle degenerate cases consistent with the perturbation. None of these steps are particularly surprising; the same ideas can be found in both ad-hoc approaches used while coding and in general perturbation schemes. What we did

find surprising was how many other goals could be achieved at the same time by a problem-specific approach.

In addition to removing degenerate cases, the implementation of an ideal perturbation scheme might have the following properties:

1. It perturbs the input geometrically so that the computed result is a valid output for an infinitesimally-perturbed problem.
2. It produces a canonical result, irrespective of the order of input data, or when other information is to be encoded in the order of the data. This is helpful for testing or verifying randomized or dynamic algorithms.
3. It adds little computational overhead. In particular, the only additional computation for non-degenerate cases should be to verify their non-degeneracy; if it can reuse some of that computation for resolving the degeneracies, so much the better.
4. It should not require more precision than the test for the existence of a degeneracy.
5. It should be able to resolve degeneracies for more than one test. Even better if it can be extended to further tests without requiring recomputation of previous results.

After reviewing perturbation schemes in the literature in the next section, we use Delaunay triangulations under polygonal and Euclidean metrics in 2-d and 3-d as our primary examples and observe that most of these goals can be achieved simultaneously. What we lose is generality; our perturbations and the resulting code for their implementation are tuned to a particular problem and set of predicates. Instead we illustrate how our three step approach can guide the search for good perturbations.

We do make one non-degeneracy assumption in this paper: that correct input contains no duplicated data. Duplicate geometric data is detected as a degeneracy by our schemes; if it is necessary to resolve these degeneracies, then the perturbation function should incorporate differences that exist in non-geometric attributes.

2 A review of perturbation schemes

The ultimate goal of a scheme that eliminates degeneracies is an implementable policy that handles all cases in a consistent manner. Because consistency depends on the algorithm, the easiest way to guarantee it is to prove that the algorithm works correctly on all non-degenerate inputs and

to choose a policy that is consistent with a perturbation of the input that removes all degenerate cases.

Since a random perturbation restores general position, some practical codes will abort a computation when they detect a degeneracy, apply a small random perturbation to all points, and solve the new problem. The perturbation's magnitude depends very much upon the problem to be solved—it must be large enough not to be lost in the numerical imprecision, but small enough that the solution on the perturbed data is somewhat related to the original. If the input data has interesting geometric structure (e.g., a triangulation), then the perturbation must not render it invalid (e.g., by reversing the orientation of a triangle).

Theoretical work usually seeks a canonical, infinitesimally-small perturbation. Several general methods have been presented in the literature. We can classify these methods based on whether they perturb the underlying numerical computations or the geometry, and whether they use additional information, or the points alone.

2.1 General perturbation schemes

Yap [Yap90] suggested that evaluations of the signs of test polynomials, which are considered to involve all inputs to an algorithm, can be replaced by a black-box tests that never return zero. He used an admissible ordering of the terms of the polynomials to determine a sign, and, for computations on points, he gave conditions under which the resulting signs are consistent with a perturbation of the points.

Neuhauser [Neu97] extended Yap's approach to rational functions and produced an implementation that included some practical ideas such as using an infinitesimal random displacement so that few cases would require involved analysis.

Edelsbrunner and Mücke [EM90] detailed their method of "Simulation of Simplicity" (SoS) for determinant computations, which involved using both point and coordinate indices (or location in memory) to construct a perturbation by powers of an infinitesimal ϵ. Specifically, point $p_i \in E^d$ is replaced by

$$p_i' = p_i + \left(\epsilon^{2^{id}}, \epsilon^{2^{id+1}}, \ldots, \epsilon^{2^{id+d-1}}\right).$$

For computing the sign of a determinant, Canny and Emiris [EC95, ECS97] proposed schemes that also use a unique integer for each point, but simplify the computation by reducing the powers of ϵ and allowing the use of arithmetic modulo a prime q larger than the largest index. Point

$p_i \in E^d$ is replaced by

$$p_i' = p_i + \epsilon \left(i \bmod q, i^2 \bmod q, \ldots, i^d \bmod q \right).$$

This approach is less general than Edelsbrunner and Mücke method. It can be generalized to problems other than computing the sign of a determinant, but applicability has to be checked for each particular problem.

Seidel [Sei94] showed that it is sufficient to find, for each n, an n point set Q_n for which the algorithm detects no degeneracies. The degeneracies from any other sequence of n points, p_1, p_2, ..., p_n will be removed by the infinitesimal perturbation $p_i' = p_i + \epsilon q_i$. From this perspective, Edelsbrunner and Mücke [EM90] and Canny and Emiris [EC95, ECS97] both give explicit constructions of point sets Q_n that have no degeneracies for polynomials in which each variable appears with degree unity, such as the orientation test determinant.

Assigning a unique integer to each point (for example, a memory address) is relatively easy in batch-processing algorithms, but can be more problematic in on-line or dynamic algorithms. Moreover, testing or verifying that an algorithm has produced a correct output can be complicated by the fact that the output depends on an unknown assignment of integers to points. Finally, it has recently been shown that the ordering of data can be used to encode other information, such as a triangulation [SvK97, DS97], for compact transmission over a network. Such schemes require a canonical way to handle degeneracies that does not depend on ordering.

It is possible to implement SoS without a unique index for each point by sorting the points under test lexicographically and using their relative ranks. Edelsbrunner, in private communication, has noted that the results will be consistent with a global lexicographical ordering. This approach can also be considered for Yap's scheme, where the points are given their relative ranks, and then terms are sorted; proving that the resulting order is admissible becomes difficult. The run-time overhead of sorting k points (and possibly many terms) for each predicate evaluation is prohibitive.

2.2 Delaunay triangulations and problem-specific perturbation schemes

Several problem-specific schemes have been applied to Delaunay triangulations because of their importance and because they require some perturbation for their definition.

Each distance measure defines a *unit circle* of all points at distance 1 from the origin: for the Euclidean metric the unit circle actually is the circle of radius 1, for the L_∞ metric it is the square with side length 2, and for a general polygonal distance function it is a convex polygon with the

origin in the interior. The *Delaunay triangulation* of a set of point sites is defined in terms of an *empty circle property*: An edge $\overline{pq}$ is included in the Delaunay triangulation if some homothet of the unit circle (a scaled and translated copy) with p and q on the boundary encloses no other points.

In general, three points determine the scale and translation parameters for a homothet—each point contributes one constraint, and retains one degree of freedom for its position on the boundary of the homothet. Thus, a degenerate case occurs whenever a homothet of the unit circle has empty interior but $k > 3$ points on the boundary. The definition of the Delaunay "triangulation" gives either an empty k-sided face, or a k-clique with all $\binom{k}{2}$ edges, depending on whether points on the boundary of a homothet are taken to be enclosed or not.

Delaunay triangulation algorithms for non-degenerate point sets in the plane can be implemented using two tests:

`CCW`(p, q, r) tests whether the triangle $\triangle pqr$ is oriented counter-clockwise—that is, whether r is to the left of the oriented line from p to q.

`InCircle`(p, q, r, s) tests whether s lies on the circle through the three points p, q, r.

For most algorithms, `InCircle`(p, q, r, s) is called only when `CCW`(p, q, r) is true.

Drysdale [Dry90] worked out the degenerate cases for the divide and conquer computation of the Delaunay triangulation for polygonal metrics, and found it necessary to use a clever bundling technique to ensure that the merge step took sub-quadratic time. Bundling is not necessary if the point set is perturbed.

It is well-known that the Delaunay triangulations are dual to Voronoi diagrams; the study of abstract Voronoi diagrams has identified primitives in addition to `InCircle`() and `CCW`(), most of which handle bisectors for the Voronoi [Kle89]. A relevant development is the fact that degenerate cases can be handled with a primitive that computes the Voronoi or Delaunay of any given set of five points [KMM93].

Mount and Saalfeld [MS88] gave a nearly-canonical way to complete the Delaunay triangulation of co-circular points using a triangulation that lexicographically maximizes the vector that lists the angles of all triangles in non-decreasing order. This triangulation is not necessarily the perturbation of a Delaunay triangulation [DS93], and is still non-canonical for point sets with symmetries, such as the vertices of a regular n-gon.

Edelsbrunner and Mücke report that implementing `InCircle`() directly under SoS [EM90] "turns out to be a real pain" because of mixed products that appear. We will see these mixed products

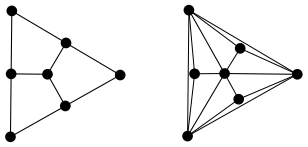


Figure 1: A triangulation that is not Delaunay, but can result from perturbing lifted points

in our perturbation, where they will, in fact, help keep the algebraic degree of the test low. Edelsbrunner and Mücke recommend that the `InCircle`() test first be transformed to an orientation test in one higher dimension by the standard lifting map, which lifts each point to a paraboloid of revolution. Then they suggest perturbing the lifted points to removed degeneracies. Using this perturbation, however, one can compute a triangulation, such as the one in Figure 1, that is not the Delaunay triangulation of any perturbed set of points in the plane. Thus, properties of the Delaunay, such as acyclicity [DFNP91], cannot be assumed for the triangulations of perturbed points.

In a completely different approach, Dillencourt and Smith [DS93] have given a simple and complete graph-theoretic characterization of the triangulations that are valid completions of a Delaunay triangulation. Using their result, the *Delaunay diagram* of a set of sites—which consists of the edges with the empty closed circle property—can be triangulated in a manner consistent with some perturbation in linear time: color the diagram with two colors if it is bipartite, and make sure that some added edge joins nodes of different colors. This is a beautiful result, but it does require computing the Delaunay diagram first; it is not suitable for canonical triangulation of dynamic or random point sets.

3 Examples of perturbing the problem and the world

We describe the technique of generating perturbations of the problem and perturbations of the world by a series of examples that progress from sorting to Delaunay triangulations with L_∞, polygonal, and Euclidean metrics, in dimensions 2 and 3. We also mention d-dimensions and arrangements of line segments.

3.1 Sorting points by x coordinate

As many computational geometers know, to sort a set of distinct points by x coordinates, one should break ties by comparing y coordinates. This policy for comparing points can be interpreted in two different, but consistent ways.

The first is as a global *perturbation of the world* that replaces $p = (p_x, p_y)$ by $p' = (p_x + \epsilon p_y,\ p_y)$. Comparing the x-coordinates of perturbed points p' and q' never produces the degenerate case: $(p_x - q_x) + \epsilon(p_y - q_y) = 0$ only if points p and q are identical.

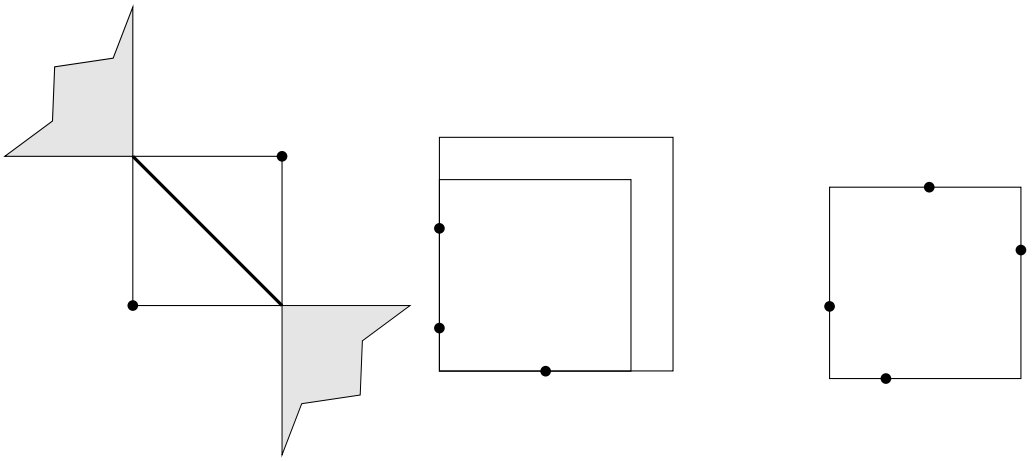
A second interpretation is as a *perturbation of the problem* that sorts the points by their projections onto the line $y = \epsilon x$ rather than onto the x axis.

This order of presentation is exactly backwards from what we will use with more complex problems. Usually, we find it easiest to conceive of a perturbation as perturbing the problem. This suggests a perturbation to apply to the world—in the case of sorting, a new axis is chosen by an infinitesimal affine transformation of the problem, and the inverse transformation is applied to all points to perturb the world. Writing the expression for the test as a polynomial in ϵ and interpreting the coefficients geometrically gives us our implementation policy: if the difference in x coordinates is zero, then compute the difference in y coordinates.

3.2 Delaunay triangulation under the L_∞ metric

Consider the L_∞ metric, in which the "unit circle" is an axis-aligned square. Degenerate cases, illustrated in Figure 2, occur whenever more than three points lie on the same square or whenever two points lie on the same side. It is sufficient to be able to test if two points have the same x coordinates or y coordinates, or if, for four points, the difference in x coordinates of the first pair equals the difference in y coordinates of the second pair. A special case of this last degeneracy occurs when two points are the vertices of an axis parallel square, and their bisector becomes a region rather than a curve.

A simple way to perturb any Delaunay problem is to change the unit circle defining the metric. Because simple perturbations lead to simple implementations, we prefer non-uniform scaling and skew transformations. Thus, we shrink the square infinitesimally in x to make sure that differences in x and y coordinates are not equal, which removes degeneracies caused by four points lying on different sides of a square. Scaling preserves horizontal and vertical lines, so we skew the x coordinates by a smaller amount to remove degeneracies caused by points with the same x coordinate, then skew the y coordinates by a still smaller amount to remove degeneracies caused



two opposite corners: degenerate bisector

non unique circle through three points

four cocircular points

Figure 2: Degenerate cases for the L_∞ metric

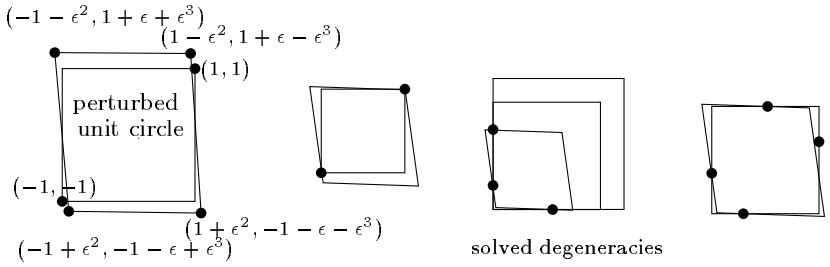


Figure 3: Perturbed L_∞ metric

by points with the same y coordinate.

Translating these changes to the metric into changes in point coordinates, we replace a point $p = (p_x, p_y)$ with

$$p' = ((1+\epsilon)p_x + \epsilon^2 p_y,\ p_y + \epsilon^3 p_x).$$

For the curious, this corresponds to perturbing the unit square by the inverse transformation into a quadrilateral whose vertices, after scaling by $1+\epsilon-\epsilon^5$, are $(1-\epsilon^2, 1+\epsilon-\epsilon^3)$, $(-1-\epsilon^2, 1+\epsilon+\epsilon^3)$, $(-1+\epsilon^2, -1-\epsilon+\epsilon^3)$, and $(1+\epsilon^2, -1-\epsilon-\epsilon^3)$

Theorem 1 proves that the perturbation of the world effectively removes all degeneracies for incrementally computing the Delaunay triangulation under the L_∞ metric.

Theorem 1 *Let P be a set of distinct points in the plane. Replacing each $p \in P$ with p', as defined above, removes all degeneracies with respect to point coordinate comparison and L_∞* `inCircle()` *tests.*

Proof We consider each of the three operations: comparison of y coordinates, of x coordinates, and of differences in x and y:

`compareY`(p, q) The difference in y coordinates between points p and q,

$$p'_y - q'_y = p_y - q_y + \epsilon^3(p_x - q_x),$$

is zero only if both y and x coordinates of p and q are identical.

`compareX`(p, q) The difference in x coordinates between points p and q,

$$p'_x - q'_x = (1 + \epsilon)(p_x - q_x) + \epsilon^2(p_y - q_y),$$

is again zero only if p and q are identical.

`compareXY`(p, q, r, s) The difference in x coordinates of p and q and y coordinates of r and s,

$$(p'_x - q'_x) - (r'_y - s'_y) = (p_x - q_x) - (r_y - s_y) + \epsilon(p_x - q_x) + \epsilon^2(p_y - q_y) + \epsilon^3(r_x - s_x),$$

is zero only if the points p and q are identical (and, as it happens, r and s are also identical).

■

The implementation of these operations are now straightforward. For `compareXY`(p, q, r, s), we return the first non-zero sign in the sequence $(p_x - q_x) - (r_y - s_y)$, $(p_x - q_x)$, $(p_y - q_y)$.

3.3 Delaunay triangulation under polygonal distance functions

We note that the technique of the previous section easily extends to resolve degeneracies under any polygonal distance function. As in the preceding section, evaluating `InCircle(p,q,r,s)` is subject to two kinds of degeneracy. The first occurs when pqr do not define an unique homothet; the second occurs when s is on the boundary of the homothet through pqr.

We illustrate by describing a sub-optimal algorithm (efficient when the distance function is defined by a constant-size convex polygon $\mathcal{C}$) to implement `InCircle`(p, q, r, s). Construct the

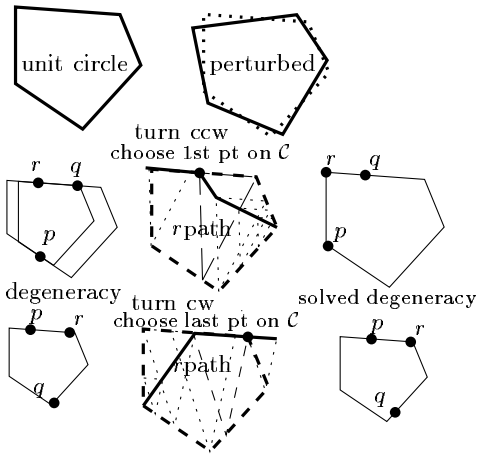


Figure 4: Perturbing convex metric $\mathcal{C}$

'circle' circumscribing p, q, r by growing a triangle homothetic to pqr inside $\mathcal{C}$, keeping contact with p and q. When point r crosses the boundary of $\mathcal{C}$ we have found the homothet of pqr inscribed in $\mathcal{C}$ or equivalently, the homothet of $\mathcal{C}$ circumscribing pqr. (The same perturbations can be applied to the search algorithm for larger $\mathcal{C}$ [KS95].)

The first degeneracy occurs if r does not cross the boundary of $\mathcal{C}$ but follows it for a while. In that case there is not a unique homothet through pqr. This degeneracy is resolved by a small rotation of angle $\arcsin(\epsilon)$ of $\mathcal{C}$. This degeneracy is easiest to treat in the "perturbing the problem" scheme. In the implementation policy, just look at the orientation used by r to follow the boundary of $\mathcal{C}$; at most one additional dot product is needed (see Figure 4). Nevertheless, we can view this a perturbing the world by the inverse rotation, replacing p with $p' = ((1-\epsilon^2)p_x - \epsilon p_y, \epsilon p_x + (1-\epsilon^2)p_y)$.

The second degeneracy is best resolved in the "perturbing the world" scheme by combining a scaling of the x coordinate with the previous perturbation. The point p becomes

$$p' = \Big((1+\epsilon)[(1-\epsilon^2)p_x - \epsilon p_y], \epsilon p_x + (1-\epsilon^2)p_y\Big) .$$

The implementation policy for this second degeneracy uses the four sides of the homothet that contain p, q, r, and s to calculate whether s goes inside or outside as the perturbation moves thes four points.

3.4 2D Delaunay triangulation under the Euclidean metric

For the Euclidean metric, we again perturb the problem by perturbing the unit circle. We scale the y coordinates to form a first ellipse; then we shear the x coordinate to create a new ellipse that cannot share more than three points with the original circle and the first ellipse. This removes degeneracies for points that are not collinear, which are the most important cases in implementing a Delaunay triangulation algorithm.

Three or more collinear points are a degenerate case for the Delaunay triangulation under any smooth metric—which includes our perturbed metric. Since affine transformations preserve collinearity, we also modify the circle in a nonlinear fashion by a perturbation that depends on the distance from the origin. The resulting perturbation of the world sends the point (p_x, p_y) to

$$p' = (p_x + \epsilon p_y,\ p_y + \epsilon^2 p_x + \epsilon^3(p_x^2 + p_y^2)).$$

Theorem 2 proves that this perturbation effectively removes the degeneracies that would affect the incremental computation of a Delaunay triangulation under the Euclidean metric.

Theorem 2 *Let P be a set of distinct points in the plane. Replacing each $p \in P$ with p', as defined above, removes all degeneracies for the Euclidean metric* `CCW()` *and* `InCircle()` *tests.*

Proof Consider first the determinant test for collinearity, `CCW`(p, q, r):

$$D = (1-\epsilon^3)\begin{vmatrix} 1 & p_x & p_y \\ 1 & q_x & q_y \\ 1 & r_x & r_y \end{vmatrix} + \epsilon^3 \begin{vmatrix} 1 & p_x & p_x^2+p_y^2 \\ 1 & q_x & q_x^2+q_y^2 \\ 1 & r_x & r_x^2+r_y^2 \end{vmatrix} + \epsilon^4 \begin{vmatrix} 1 & p_y & p_x^2+p_y^2 \\ 1 & q_y & q_x^2+q_y^2 \\ 1 & r_y & r_x^2+r_y^2 \end{vmatrix}$$

If p, q, and r are not collinear, then the sign of D is given by the sign of the first determinant as $\epsilon \to 0$.

When p, q, and r are collinear, the first determinant is zero; we can show that the second and third cannot also be zero. Consider lifting the three points to the paraboloid: $p^* = (p_x, p_y, p_x^2+p_y^2)$, $q^* = (q_x, q_y, q_x^2+q_y^2)$, and $r^* = (r_x, r_y, r_x^2+r_y^2)$. If p, q, and r are distinct, then $p^*q^*r^*$ must define a plane in 3D, since three points on the paraboloid cannot be collinear. If the first determinant is zero, then the plane $p^*q^*r^*$ is vertical; if the second is zero, the plane $p^*q^*r^*$ is parallel to the y-axis; and if the third is zero, the plane $p^*q^*r^*$ is parallel to the x-axis. It is clearly impossible for a plane to be parallel to all three axes, thus D cannot be zero unless two points are identical.

An algebraic explanation of the collinear case can also be given. If the line is not vertical, then it has an equation $y = mx + b$, so that we could write the second determinant as the determinant of a Vandermonde matrix:

$$\begin{vmatrix} 1 & p_x & p_x^2+p_y^2 \\ 1 & q_x & q_x^2+q_y^2 \\ 1 & r_x & r_x^2+r_y^2 \end{vmatrix} = (m^2+1) \begin{vmatrix} 1 & p_x & p_x^2 \\ 1 & q_x & q_x^2 \\ 1 & r_x & r_x^2 \end{vmatrix} = (m^2+1)(p_x - q_x)(q_x - r_x)(r_x - p_x).$$

This determinant is zero only if two points have the same x coordinate, which, since they lie on a line that is not parallel to the y-axis, implies that they are identical. When the line is parallel to the y-axis, all x coordinates are equal and the second determinant is zero. The third can then be written as a Vandermonde matrix:

$$\begin{vmatrix} 1 & p_y & p_x^2+p_y^2 \\ 1 & q_y & q_x^2+q_y^2 \\ 1 & r_y & r_x^2+r_y^2 \end{vmatrix} = \begin{vmatrix} 1 & p_y & p_y^2 \\ 1 & q_y & q_y^2 \\ 1 & r_y & r_y^2 \end{vmatrix} = (p_y - q_y)(q_y - r_y)(r_y - p_y).$$

Again, the determinant is zero only if two points are identical.

Notice that the sign of a Vandermonde matrix can be calculated by multiplying the signs of the terms, so adding the non-linear terms to the determinant does not necessarily raise the algebraic degree of the test.

Next, consider `InCircle`(p, q, r, s), which tests the sign of determinant

$$D = \begin{vmatrix} 1 & p'_x & p'_y & p'^2_x+p'^2_y \\ 1 & q'_x & q'_y & q'^2_x+q'^2_y \\ 1 & r'_x & r'_y & r'^2_x+r'^2_y \\ 1 & s'_x & s'_y & s'^2_x+s'^2_y \end{vmatrix}$$

The sign of D locates s with respect to the circle passing through p, q and r. Again, we expand D as a polynomial in ϵ and look at the coefficients of the lowest powers, which determine the sign as $\epsilon \to 0$.

$$D = \begin{vmatrix} 1 & p_x & p_y & p_x^2+p_y^2 \\ 1 & q_x & q_y & q_x^2+q_y^2 \\ 1 & r_x & r_y & r_x^2+r_y^2 \\ 1 & s_x & s_y & s_x^2+s_y^2 \end{vmatrix} + 2\epsilon \begin{vmatrix} 1 & p_x & p_y & p_xp_y \\ 1 & q_x & q_y & q_xq_y \\ 1 & r_x & r_y & r_xr_y \\ 1 & s_x & s_y & s_xs_y \end{vmatrix} + \epsilon^2 \begin{vmatrix} 1 & p_x & p_y & p_y^2 \\ 1 & q_x & q_y & q_y^2 \\ 1 & r_x & r_y & r_y^2 \\ 1 & s_x & s_y & s_y^2 \end{vmatrix} + 2\epsilon^2 \begin{vmatrix} 1 & p_x & p_y & p_xp_y \\ 1 & q_x & q_y & q_xq_y \\ 1 & r_x & r_y & r_xr_y \\ 1 & s_x & s_y & s_xs_y \end{vmatrix} + \epsilon^3 D',$$

where D' is a polynomial in ϵ that we expand later if the determinants that are the coefficients of the ϵ^0, ϵ^1, and ϵ^2 terms are zero.

Notice that the second and fourth determinant are equal, and that the three first determinants can be evaluated quickly if we expand by minors in the fourth column. Using a generic fourth column $(PQRS)$, we want to know when

$$\begin{vmatrix} 1 & q_x & q_y \\ 1 & r_x & r_y \\ 1 & s_x & s_y \end{vmatrix} P - \begin{vmatrix} 1 & p_x & p_y \\ 1 & r_x & r_y \\ 1 & s_x & s_y \end{vmatrix} Q + \begin{vmatrix} 1 & p_x & p_y \\ 1 & q_x & q_y \\ 1 & s_x & s_y \end{vmatrix} R - \begin{vmatrix} 1 & p_x & p_y \\ 1 & q_x & q_y \\ 1 & r_x & r_y \end{vmatrix} S = 0 \qquad (1)$$

Let us take a brief digression to study equation 1. Note that the 3×3 determinants are standard `CCW`() tests. Unless p, q, r, and s are all collinear, at least one of these determinants is non-zero; we assume that s does not lie on the line through p and q.

If equation 1 is satisfied when (P, Q, R, S) is $(p \cdot p, q \cdot q, r \cdot r, s \cdot s)$, then there are scalars α, β, γ such that $\alpha + \beta x + \gamma y = x^2 + y^2$ for the coordinates of each of the four points p, q, r, and s—that is, the four points are co-circular. If equation 1 is also satisfied for $(p_x p_y, q_x q_y, r_x r_y, s_x s_y)$ and $(p_y^2, q_y^2, r_y^2, s_y^2)$, then there are scalars such that $\alpha' + \beta' x + \gamma' y = xy$ and $\alpha'' + \beta'' x + \gamma'' y = y^2$.

If we now consider any conic passing through p, q and r. Combining all the preceding equations and the equation of the conic, we get that s should also belong to the conic. However, there is a degenerate conic consisting of the line $\overline{pq}$ and any line through r that does not include s. Since we assumed that s does not lie on the line through p and q, we have a contradiction. Therefore, either the above determinants are not all equal to zero, or the four points are collinear.

Note that, from the point of view of incremental Delaunay triangulation, we are done. The main use of the `InCircle` test is to locate a query point with respect to the circle circumscribing a triangle of the Delaunay triangulation of a set of points $\mathcal{S}$. The only triangles with collinear points, therefore, are those on the convex hull of the point set, and if a bounding triangle or box (or point at infinity) is used to begin the triangulation, then the `InCircle` test cannot be called with four collinear points.

However, the degeneracy is resolved even in the collinear case. The polynomial expression for D must be expanded further, but many of the coefficients will be zero. As before, we distinguish two cases, depending on whether the line is parallel to the y-axis.

- If the points all lie on a line parallel to the y-axis, then $p_x = q_x = r_x = s_x$ and all terms

x or x^2 disappear from the expansion of D by collinearity with the first column.

$$D = \begin{vmatrix} 1 & \epsilon p_y & p_y + \epsilon^3 p_y^2 & p' \cdot p' \\ 1 & \epsilon q_y & q_y + \epsilon^3 q_y^2 & q' \cdot q' \\ 1 & \epsilon r_y & r_y + \epsilon^3 r_y^2 & r' \cdot r' \\ 1 & \epsilon s_y & s_y + \epsilon^3 s_y^2 & s' \cdot s' \end{vmatrix} = \epsilon^4 \begin{vmatrix} 1 & p_y & p_y^2 & p' \cdot p' \\ 1 & q_y & q_y^2 & q' \cdot q' \\ 1 & r_y & r_y^2 & r' \cdot r' \\ 1 & s_y & s_y^2 & s' \cdot s' \end{vmatrix}$$

Now, when expanding the last column, all terms in y with degree at most two can be removed as proportional to one of the three first columns.

$$D = \epsilon^4 \begin{vmatrix} 1 & p_y & p_y^2 & 2\epsilon^3 p_y^3 + \epsilon^6 p_y^4 \\ 1 & q_y & q_y^2 & 2\epsilon^4 q_y^3 + \epsilon^6 q_y^4 \\ 1 & r_y & r_y^2 & 2\epsilon^4 r_y^3 + \epsilon^6 r_y^4 \\ 1 & s_y & s_y^2 & 2\epsilon^4 s_y^3 + \epsilon^6 s_y^4 \end{vmatrix} = 4\epsilon^7 \begin{vmatrix} 1 & p_y & p_y^2 & p_y^3 \\ 1 & q_y & q_y^2 & q_y^3 \\ 1 & r_y & r_y^2 & r_y^3 \\ 1 & s_y & s_y^2 & s_y^3 \end{vmatrix} + \epsilon^{10} D'_x$$

This determinant is of a Vandermonde matrix, that is non-zero unless two points share the same y coordinates. Since we have assumed that the points are distinct, and we know that they share x coordinates, the determinant is non-zero.

- Otherwise the points lie on a line with equation $y = mx + b$. In a manner similar to the previous case, we may use this expression for y and remove all terms proportional to other columns of the determinant to obtain another Vandermonde matrix:

$$D = \epsilon^6 (1 + \epsilon m)(m + \epsilon^2)(1 + m^2)^2 \begin{vmatrix} 1 & p_x & p_x^2 & p_x^3 \\ 1 & q_x & q_x^2 & q_x^3 \\ 1 & r_x & r_x^2 & r_x^3 \\ 1 & s_x & s_x^2 & s_x^3 \end{vmatrix} + \epsilon^9 D'_{xy}.$$

This makes it easy to determine the sign of D by comparing coordinates: The value of the determinant is the product of all differences in x coordinates, and the slope $m = (p_y - s_y + \epsilon)/(p_x - s_x)$, where the ϵ makes difference in y coordinates positive if it would be zero. The final sign is

$$\text{sign}(D) = ((p_y - s_y + \epsilon)(p_x - q_x)(p_x - r_x)(q_x - r_x)(q_x - s_x)(r_x - s_x)).$$

■

In fact, a careful analysis of this case shows that, since all points lie on a line, only three terms are required: $\text{sign}(D) = \text{sign}((p_y - q_y)(q_x - s_x)(r_x - s_x))$.

Implementation remark: Notice that, in the case of the 2-d Euclidean Delaunay triangulation, the tests on perturbed points can be implemented efficiently and without raising the algebraic degree. For example, the standard `InCircle()` determinant can be evaluated by minors of the last column. In the case of co-circular but not collinear points, this expansion is reused, changing the last column to the square of ordinates and, if necessary, to the product of abscissae and ordinates. The collinear case is detected when the minors are all zero, and the degeneracy is resolved by comparison of point coordinates.

3.5 Three dimensional Euclidean Delaunay

Visualizing a valid perturbation of the problem in higher dimensions is, of course, more difficult. It can help to construct a perturbation in stages. We illustrate the process for the three-dimensional Euclidean Delaunay: using low-order perturbation terms to resolve low order degeneracies, and then adding on higher order terms as necessary. This can also be a useful technique if there are only certain degeneracies that need to be handled by an algorithm.

We will begin by constructing a perturbation for the `InSphere`(p, q, r, s, t) test that handles the case of 5 points on a finite-radius sphere, which is the most important degenerate case in incremental Delaunay triangulation. Then we will add terms to handle coplanar, but not collinear, points in both the `Orient`(p, q, r, s) and `InSphere()` tests. Finally, we handle collinear points in both tests. We point out that the implementation policy once again will detect which type of degeneracy occurs as it evaluates the first determinant, and then handle the appropriate case. It does not need to test each case in turn.

We start by skewing along each of the coordinate axes, as in the plane. We use a perturbation that sends the point (p_x, p_y, p_z) to

$$p' = (p_x + \epsilon p_y, p_y + \epsilon^3 p_z, p_z + \epsilon^4 p_x),$$

The powers on the ϵs are chosen by first using a different ϵ for each skew, then choosing exponents that make the proof easiest.

Theorem 3 *Let P be a set of points in 3-D, no four of which are co-planar. Replacing each $p \in P$ with p', as defined above, removes all co-spherical 5-tuples.*

Proof The test `InSphere`(p, q, r, s, t) evaluates the sign of determinant

$$D = \begin{vmatrix} 1 & p'_x & p'_y & p'_z & p' \cdot p' \\ 1 & q'_x & q'_y & q'_z & q' \cdot q' \\ 1 & r'_x & r'_y & r'_z & r' \cdot r' \\ 1 & s'_x & s'_y & s'_z & s' \cdot s' \\ 1 & t'_x & t'_y & t'_z & t' \cdot t' \end{vmatrix}$$

The sign of D locates t with respect to the sphere passing through p, q, r and s. We will abbreviate 5×5 determinants of this form by listing the variable for each row, followed by a pattern for all rows: e.g.,

$$D = \mathrm{Det}[p', q', r', s', t'; 1, x, y, z, (x^2 + y^2 + z^2)]$$

Again, we expand D as a polynomial in ϵ and look at the coefficients of the lowest powers, which determine the sign as $\epsilon \to 0$.

$$\begin{aligned} D &= (1 + \epsilon^8)\mathrm{Det}[p, q, r, s, t; 1, x, y, z, (x^2 + y^2 + z^2)] \\ &\quad + 2\epsilon\mathrm{Det}[p, q, r, s, t; 1, x, y, z, xy] + \epsilon^2\mathrm{Det}[p, q, r, s, t; 1, x, y, z, y^2] \\ &\quad + 2\epsilon^3\mathrm{Det}[p, q, r, s, t; 1, x, y, z, yz] + 2\epsilon^4\mathrm{Det}[p, q, r, s, t; 1, x, y, z, xz] \\ &\quad + \epsilon^6\mathrm{Det}[p, q, r, s, t; 1, x, y, z, z^2] + \epsilon^8\mathrm{Det}[p, q, r, s, t; 1, x, y, z, x^2] + \epsilon^9 D' \end{aligned}$$

where D' is a polynomial in ϵ that we expand later if the determinants that are the coefficients of the terms in ϵ with degree smaller than 8 all evaluate to zero.

Similarly to the 2D case, if all determinants are zero, then we conclude that any quadric passing through four of the points has to include the fifth. If, without loss of generality, t is not in the plane passing through p, q and r, then it is possible to find a degenerate quadric consisting of this plane and another plane that contains s and not t, which yields a contradiction. Thus the determinants are all zero only if the five points are coplanar.

As in the plane, if the `InSphere` test is used to locate t with respect to a non-degenerate tetrahedron, we can guarantee that the test will always conclude at that point. Notice that the sum of the third, sixth, and last determinants is equal to the determinant corresponding to the non-perturbed `InSphere` test, so the last determinant never needs to be evaluated. Thus in the non-coplanar case, the `InSphere` test needs at most 6 determinant evaluations, all having dimension and degree 5 and sharing 4 columns in common. ■

Since affine transformations preserve co-planarity, our perturbation has not removed degeneracies for the `Orient`(p, q, r, s) test. We add non-linear terms to our perturbation, and send the point (p_x, p_y, p_z) to

$$p' = (p_x + \epsilon p_y + \epsilon^9(p_x^2 + p_y^2 + p_z^2), p_y + \epsilon^3 p_z + \epsilon^{10}(p_x^2 + p_y^2 + p_z^2), p_z + \epsilon^4 p_x + \epsilon^{11}(p_x^2 + p_y^2 + p_z^2))$$

Theorem 4 *Let P be a set of points in 3-D, no three of which are collinear. Replacing each $p \in P$ with p', as defined above, removes all co-spherical 5-tuples and co-planar 4-tuples.*

Proof The orientation test `Orient`(p, q, r, s) evaluates the sign of determinant

$$D = \begin{vmatrix} 1 & p'_x & p'_y & p'_z \\ 1 & q'_x & q'_y & q'_z \\ 1 & r'_x & r'_y & r'_z \\ 1 & s'_x & s'_y & s'_z \end{vmatrix}$$

If the four original points are not coplanar, the sign of D is the sign of the usual (unprimed) orientation determinant. Otherwise we can rewrite D as a polynomial in ϵ by expanding columns and eliminating all linear perturbations, keeping only quadratic ones:

$$D = (\epsilon^9 - \epsilon^{11}\begin{vmatrix} 1 & p \cdot p & p_y & p_z \\ 1 & q \cdot q & q_y & q_z \\ 1 & r \cdot r & r_y & r_z \\ 1 & s \cdot s & s_y & s_z \end{vmatrix} + \epsilon^{10}\begin{vmatrix} 1 & p_x & p \cdot p & p_z \\ 1 & q_x & q \cdot q & q_z \\ 1 & r_x & r \cdot r & r_z \\ 1 & s_x & s \cdot s & s_z \end{vmatrix} + \epsilon^{11}\begin{vmatrix} 1 & p_x & p_y & p \cdot p \\ 1 & q_x & q_y & q \cdot q \\ 1 & r_x & r_y & r \cdot r \\ 1 & s_x & s_y & s \cdot s \end{vmatrix}$$

Considering usual lifting on the 4D paraboloid, we get four points $p^\star$, $q^\star$ $r^\star$ and $s^\star$. Assume that p, q, r and s are not collinear, then $p^\star q^\star r^\star s^\star$ must define an hyper-plane in 4D. If all the determinants are zero, we get that this hyper-plane must be parallel to all the four axis of coordinates in 4D, which is clearly impossible, thus D cannot be 0 unless than the four points p, q, r and s are collinear. Unlike the 2D case, these matrices are not Vandermonde and thus the algebraic degree of the tests is raised by the perturbation to the degree of the `InSphere`() test.

By the way, this perturbation also handles the degenerate cases of `InSphere`(p, q, r, s, t) in which not all points are collinear. Since the powers of ϵ are chosen large enough on the new perturbation terms, the proof of Theorem 3 handles all but the co-planar cases.

We introduce several vectors in five dimensions corresponding to the columns of some determinants. We denote the vectors $(1,1,1,1,1)$, $(p_x, q_x, r_x, s_x, t_x)$ or $(p_x^2, q_x^2, r_x^2, s_x^2, t_x^2)$ by V_1, V_x or V_{x^2}, respectively. We also denote $V_{x^2} + V_{y^2} + V_{z^2}$ by V_{R^2}.

The co-planarity of the five points can be interpreted as the fact that the dimension of the vector space $F_1 = Vect(V_1, V_x, V_y, V_z)$ is three. Assume that the dimension of $F_2 = Vect(V_1, V_x, V_y, V_z, V_{R^2})$ is four.

Then the next terms in the development of D as a polynomial in ϵ give other vectors in five dimension. The coefficient of ϵ^{10} is

$$\mathrm{Det}[p, q, r, s, t; 1, x, y, x^2 + y^2 + z^2, xy].$$

If this term is equal to zero, then we deduce that the vector $V_{xy} \in F_2$. Similarly, coefficient of ϵ^{11} yields that vector $V_{y^2} \in F_2$ Continuing the development, either we find a non-zero determinant, or we conclude that $V_{x^2}, V_{y^2}, V_{z^2}, V_{xy}, V_{yz}, V_{xz} \in F_2$. We assume, without loss of generality, that $F_2 = Vect(V_1, V_x, V_y, V_{x^2})$.

Now, the next term in the development of D is the coefficient of ϵ^{18} is

$$\mathrm{Det}[p, q, r, s, t; 1, x, y, (x^2 + y^2 + z^2), x(x^2 + y^2 + z^2)].$$

This determinant is zero, if and only if $V_{xR^2} \in F_2$. But, $V_{xR^2} = V_{x^3} + V_{xy^2} + V_{xz^2}$ and since $V_{y^2}, V_{z^2} \in F_2 = Vect(V_1, V_x, V_y, V_{x^2})$ we can rewrite V_{y^2} and V_{z^2} as a linear combination of V_1, V_x, V_y, V_{x^2} and thus rewrite V_{xy^2} and V_{xz^2} as a linear combination of $V_x, V_{x^2}, V_{xy}, V_{y^2}$. If we continue this rewriting scheme, we find that $V_{xR^2} \in Vect(V_1, V_x, V_y, V_{x^2}, V_{x^3})$. If the determinant is zero, then we have $F_2 = Vect(V_1, V_x, V_{x^2}, V_{x^3})$, and the next coefficient in D can be rewritten as a Vandermonde determinant

$$\begin{vmatrix} 1 & p_x & p_x^2 & p_x^3 & p_x^4 \\ 1 & q_x & q_x^2 & q_x^3 & q_x^4 \\ 1 & r_x & r_x^2 & r_x^3 & r_x^4 \\ 1 & s_x & s_x^2 & s_x^3 & s_x^4 \\ 1 & t_x & t_x^2 & t_x^3 & t_x^4 \end{vmatrix}$$

which is zero if two points have the same x coordinate. Continuing the process, we get Vandermonde determinants with respect to other coordinates, and conclude that the plane containing the points should include lines parallel to the three axes which is impossible.

Solving degeneracies in that manner needs the use of predicates of degree up to 7 although the original predicate is of degree 5. ■

This process can be continued to the collinear cases if we consider the perturbation

$$p' = (p_x + \epsilon p_y + \epsilon^9 (p \cdot p),\ p_y + \epsilon^3 p_z + \epsilon^{10}(p \cdot p),\ p_z + \epsilon^4 p_x + \epsilon^{11}(p \cdot p) + \epsilon^{30}(p_x^3 + p_y^3 + p_z^3)).$$

Theorem 5 *Let P be a set of points in 3-D. Replacing each $p \in P$ with p', as defined above, removes all co-spherical 5-tuples, co-planar 4-tuples, and collinear triples.*

We leave this proof to the algebraically-inclined reader. Suffice it to say that we need consider only the collinear case since the power of ϵ on the new perturbation term was chosen so as not to affect the results of Theorem 4. For one example, To evaluate the `Orient`() test for points on a line not parallel to a coordinate plane, one finds that ϵ^{39} is the first term term that might not be zero, and for distinct points it is not zero, since it reduces to a Vandermonde determinant by row operations.

Implementation remark: Notice that, as in two dimensions, these predicates can be implemented efficiently. For one thing, a test can be evaluated in the same order as it was constructed. This can lead to a significant savings in the worst-case number of terms that must be evaluated, compared with general perturbation schemes such as SoS. For example, the 3-d test for `Orient`() is a polynomial in ϵ with 99 terms, but only evaluates the orientation determinant, then three more determinants of Theorem 4 if the points are co-planar, and one more if the points are collinear (the coefficient on ϵ^{39}).

Also as in two dimension, portions of the computation can be reused. For example, expansion of the `InCircle`() determinant by minors in the last column implies that evaluation of 5 minors followed by at most 6 dot products solves the non-coplanar cases. Co-planar cases are solved using up to 6 additional degree 7 determinants, or three VanderMonde determinants.

3.6 Arrangements of Segments

We may illustrate the possibility of selecting which degeneracies to resolve on the problem of the arrangement of line segments. An arrangement of line segments can present several kinds of degeneracies:

1. end points or intersection points with same x,
2. more than two segments crossing at the same intersection point,
3. overlapping segments on the same line,
4. more than one segment ending at the same end point,

We may be interested in removing some kinds of degeneracies and not the others.

A small rotation or skew will resolve only the first kind of degeneracies. Perturbation with a quadratic term (similar to one used for Euclidean Delaunay) will resolve the first three kinds of

degeneracies, while a perturbation that uses one end-point of the segment to perturb the other will resolve all degeneracies. Neuhauser [Neu97] developed this idea for perturbations that assign unique integers to the input.

4 Conclusion

We have shown that problem-specific schemes for resolving degeneracies can be developed by a combination of *perturbing the problem* and *perturbing the world*. In both techniques, a parameter ϵ is introduced and when ϵ goes to zero, the limit case is our original problem. The *perturbing the problem* approach transforms the definition of the problem while the *perturbing the world* approach introduces a global transformation of the world of data.

In low dimensional computational geometry, these techniques are simple to conceive and to implement. They also provide a way to resolve degeneracies without a numbering of a data.

The main advantages of these techniques are

- to help to conceive a perturbation for a particular purposes
- to yield an efficient implementation
- to give canonical output (independent from a numbering of the points)
- to allow geometric control of that output (for example, replacing convex hull by an α-shape $\alpha \to 0$ will gives all the points on the boundary of the convex hull or only the vertices depending of the sign of α).

For the Delaunay triangulation, we show that our techniques provides efficient `InCircle` and `InSphere` tests for polygonal metrics and, in the Euclidean case, for a circle or sphere with finite radius. By efficient, we mean that in case of degeneracy the additional computations is small and does not increase the precision required to make the computation. The degeneracies for orienting a triangle, or the `InCircle` test on four collinear points are also resolved efficiently. The perturbation scheme allows the resolution of all degeneracies, including those not used by most algorithms that construct Delaunay triangulation. For resolving second order degeneracies in 3D, the arithmetic degree of additional tests does increase.

5 Acknowledgement

We thank the anonymous referee for corrections and important comments, and for pointing out the work of Neuhauser [Neu97].

References

[DFNP91] L. De Floriani, B. Falcidieno, G. Nagy, and C. Pienovi. On sorting triangles in a Delaunay tessellation. *Algorithmica*, 6:522–532, 1991.

[Dry90] R. L. Drysdale, III. A practical algorithm for computing the Delaunay triangulation for convex distance functions. In *Proc. 1st ACM-SIAM Sympos. Discrete Algorithms*, pages 159–168, 1990.

[DS93] M. B. Dillencourt and W. D. Smith. A simple method for resolving degeneracies in Delaunay triangulation. In *Proc. 20th Internat. Colloq. Automata Lang. Program.*, volume 700 of *Lecture Notes Comput. Sci.*, pages 177–188. Springer-Verlag, July 1993.

[DS97] Markus Denny and Christian Sohler. Encoding a triangulation as a permutation of its point set. In *Canadian Conference on Computational Geometry*, pages 39–43, August 1997.

[EC95] I. Emiris and J. Canny. A general approach to removing degeneracies. *SIAM J. Comput.*, 24:650–664, 1995.

[ECS97] I. Z. Emiris, J. F. Canny, and R. Seidel. Efficient perturbations for handling geometric degeneracies. *Algorithmica*, 19(1–2):219–242, September 1997.

[EM90] 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.

[Kle89] R. Klein. *Concrete and Abstract Voronoi Diagrams*, volume 400 of *Lecture Notes Comput. Sci.* Springer-Verlag, 1989.

[KMM93] R. Klein, K. Mehlhorn, and S. Meiser. Randomized incremental construction of abstract Voronoi diagrams. *Comput. Geom. Theory Appl.*, 3(3):157–184, 1993.

[KS95] D. Kirkpatrick and J. Snoeyink. Tentative prune-and-search for computing fixed-points with applications to geometric computation. *Fundam. Inform.*, 22:353–370, 1995.

[MS88] D. M. Mount and A. Saalfeld. Globally-equiangular triangulations of co-circular points in $O(n \log n)$ time. In *Proc. 4th Annu. ACM Sympos. Comput. Geom.*, pages 143–152, 1988.

[Neu97] M. A. Neuhauser. Symbolic perturbations and special arithmetics for controlled handling of geometric degeneracies. In *Proc 5th WSCG'97*, pages 386–395, Plzen, Czech Republic, 1997.

[Sei94] R. Seidel. The nature and meaning of perturbations in geometric computing. In *Proc. 11th Sympos. Theoret. Aspects Comput. Sci.*, volume 775 of *Lecture Notes Comput. Sci.*, pages 3–17. Springer-Verlag, 1994.

[SvK97] J. Snoeyink and M. van Kreveld. Good orders for incremental (re)constructions. In *Proc. 13th Annu. ACM Sympos. Comput. Geom.*, pages 400–402, 1997.

[Yap90] C. K. Yap. A geometric consistency theorem for a symbolic perturbation scheme. *J. Comput. Syst. Sci.*, 40(1):2–18, 1990.