A Reliable Algorithm for Computing the Generalized Voronoi Diagram for a Set of Spheres in the Euclidean d-dimensional Space

M. L. Gavrilova

Dept of Comp. Science, University of Calgary
Calgary, AB, Canada, T2N1N4
marina@cpsc.ucalgary.ca

Abstract. We present a new algorithm for reliable computation of the Euclidean Voronoi diagram vertex for a set of spheres in a fixed length floating-point arithmetic. The algorithm is provided for the d-dimensional case and is implemented for the 3-dimensional Voronoi diagram of a set of spheres.

Keywords: Reliable computing, fixed precision floating-point arithmetic, Euclidean Voronoi Diagram, d-dimensional space, interval arithmetic

1 Introduction

In areas such as motion planning, computer simulation of physical systems, mechanical engineering, robotics and computer graphics, the focus of study is often on the geometric objects, approximated by spheres. In these applications, the issues concerning robustness and numerical stability of algorithms, as well as the actual running times of their implementations, more and more often become centers of attention [1, 3, 4, 12].

Numerical degeneracies, resulting from performing calculations on the real numbers in a computer floating-point arithmetic, are often handled by approximation [11]. Over the last 25 years, an alternative approach to real computation has been developed, called *exact real computation* in which results are generated exactly [2, 5, 7, 8, 12]. As identified by L. Blum [1], the state of the art in development of numerical and simulation software, as well as computer algebra systems, is moving towards exact systems which could be used to replace the traditional approximate systems.

The main contribution of this paper is in a new algorithm to compute the generalized Voronoi diagram vertex in R^d with the precision to the smallest representable floating-point number in a given computer architecture. Note, that the computation of the exact Delaunay triangulation [2, 6] is a much simpler task, which does not involve computing the d-dimensional coordinates of the Voronoi vertices. It is not feasible to use a traditional approach of computing the Voronoi vertices using libraries for exact computation, such as PRECISE [7] or LEDA [8], not only due to the loss of efficiency (see similar studies on exact segment intersection testing [5]), but also due to inability of those libraries to compute the square root from a floating-point number in a fixed precision arithmetic.

To tackle the problem, we developed an original approach, which combines an interval analysis, the ESSA method for computing the sign of algebraic equation exactly, a problem reduction technique for a system of $(d+1)$ quadratic algebraic equations and an iterative approximation method to compute the intersection of d-dimensional quasi-halfspaces with hyperbolic boundaries. The algorithms is formulated for the d-dimensional case and the proof of its correctness is provided. The algorithm is implemented for a set of spheres in 3D.

2 The Problem

Given $(d+1)$ spheres in R^d: $P_i = \{\mathbf{p}_i = (x_{i1}, x_{i2}, ..., x_{id}), r_i\}, i = 1..d+1$, where x_{ij} and r_i are represented by floating-point numbers. The Voronoi diagram is used to store the topological information about the system of spheres as well as to answer various queries.

Definition 1. A generalized Euclidean Voronoi diagram for a set of spheres S in R^d is the set of Voronoi regions $\left\{\mathbf{x} \in R^d \,\middle|\, d(\mathbf{x},P) \leq d(\mathbf{x},Q), \forall Q \in S - \{P\}\right\}$, where $d(\mathbf{x},P)$ is the Euclidean distance function between a point $\mathbf{x}$ and a sphere $P \in S$.

The distance between a point $\mathbf{x}$ and a sphere P with center at $\mathbf{p}$ and radius r_p is defined as

$$d(\mathbf{x},P) = d(\mathbf{x},\mathbf{p}) - r_p, \tag{1}$$

where

$$d(\mathbf{x},\mathbf{p}) = \sqrt{\sum_{i=1}^{d} (x_i - p_i)^2}. \tag{2}$$

According to the definition, the generalized Voronoi vertex is obtained as the intersection of d quasi-halfspaces with hyperbolic boundaries (see Fig. 1). The task is to find the best floating-point approximation of the Voronoi vertex, i.e.

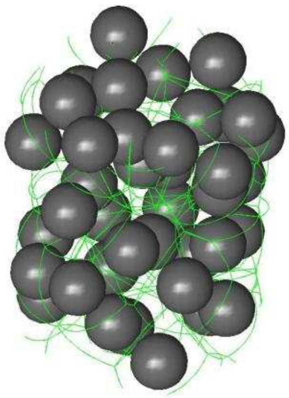

Figure 1: The Euclidean Voronoi diagram of a set of spheres in 3D

the point in the space equidistant from these spheres.

3 The tools

3.1 The ESSA Algorithm

The ESSA algorithm computes the sign of an algebraic sum exactly using standard fixed precision floating-point arithmetic [9]. The summands are represented as normalized single precision floating-point numbers with fixed mantissa length. The main idea of the algorithm is to separate the positive and the negative summands into two lists, sort the two lists separately and then decimate leading terms recursively. More details on algorithm implementation can be found in [6]. Applying series of transformations, a product of single precision floating-point numbers can be converted to a sum of single-precision floating-point numbers. Thus, the ESSA can be used to compute the sign of any algebraic expression, containing sums and products, exactly.

3.2 Interval Analysis

Interval analysis is a tool for including numerical computations. This means that a function $f(x)$ defined using the standard arithmetic operations $(+,-,\times,/$ which are the only arithmetic operations available on a computer) can be evaluated using interval arithmetic obtaining an interval $F(x)$ which has the property that the inclusion $f(x) \in F(x)$ is guaranteed. More on interval arithmetic can be found in [10].

4 The Analytical Solution

In this section, we first obtain the algebraic representation of the solution for the coordinates and the radius of the inscribed sphere (i.e. the Voronoi vertex) in R^d. Then, we establish the obtain an estimate of the number of inscribed spheres in the Euclidean metric, which is required to derive the algorithm for computation of the best floating point approximation of a Euclidean Voronoi vertex in d dimensions.

Definition 2. A sphere $C = \{\xi, \rho\}$ inscribed among $d+1$ spheres $P_1, P_2, ..., P_{d+1}$ is a sphere with center $\xi = (\xi_1, \xi_2, ..., \xi_d)$ and radius ρ, such that $\rho = d(\xi, P_1) = d(\xi, P_2) = ... = d(\xi, P_{d+1})$.

The center of the sphere represents the vertex of the Voronoi diagram. First, we apply the reduction technique to obtain a system of d linear equations with $d+1$ unknown variables $(\xi_1, \xi_2, ..., \xi_d, \rho)$. This system will be then transformed to obtain a quadratic equation of ρ. Quadratic roots will be found precisely using the iterative approximation method, from which the coordinates of the Voronoi vertex will be derived.

In order to obtain the system of linear equations, let us reduce radii of the $d+1$ spheres by the radius of the smallest sphere (assume that it is r_{d+1}). Define a coordinate system with the center of coordinates at $\mathbf{p}_{d+1}$. Coordinates of the spheres after the transformation are:

$$P_i^* = \{\mathbf{p}_i^* = (x_{i1}^*, x_{i2}^*, ..., x_{id}^*), r_i^*\}, i = 1..d, \tag{3}$$

where $x_{ij}^* = x_{ij} - x_{d+1,j}, i, j = 1..d$, and $r_i^* = r_i - r_{d+1}$.

The last sphere is transformed into a point at the origin of coordinates. The coordinates of the center of the inscribed sphere C change to $\xi_j^* = \xi_j - x_{d+1,j}, j = 1..d$, and the new radius is $\rho^* = \rho + r_{d+1}$.

In Euclidean metric, coordinates of the inscribed sphere satisfy following system of equations:

$$d(\xi^*, \mathbf{p}_i^*) = \rho^* + r_i^*, i = 1..d+1. \tag{4}$$

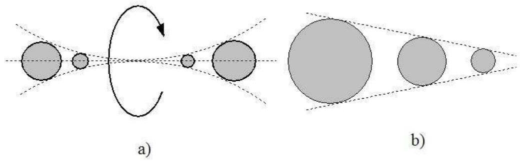

Figure 2: a) Infinite number of inscribed spheres; b) No inscribed spheres

Expanding the distance function, we obtain:

$$\begin{cases} (x_{i1}^* - \xi_1^*)^2 + ... + (x_{id}^* - \xi_d^*)^2 = (\rho^* + r_i^*)^2 \\ (\xi_1^*)^2 + (\xi_2^*)^2 + ... + (\xi_d^*)^2 = (\rho^*)^2 \end{cases} \tag{5}$$

where $i = 1..d$. The last equation can be subtracted from the remaining equations to cancel the quadratic terms. Thus, we arrive at:

$$\begin{cases} x_{i1}^*\xi_1^* + x_{i2}^*\xi_2^* + ... + x_{id}^*\xi_d^* + \rho^* r_i^* = w_i^* \\ (\xi_1^*)^2 + (\xi_2^*)^2 + ... + (\xi_d^*)^2 = (\rho^*)^2 \end{cases} \tag{6}$$

where $w_i^* = \frac{1}{2}(x_{i1}^*)^2 + (x_{i2}^*)^2 + ... + (x_{id}^*)^2 - (r_i^*)^2, i = 1..d$.
The first d equations are linear in $(\xi_1, \xi_2 ..., \xi_d, \rho)$. Denote the matrix of the system by $\mathbf{A_r}$, the column of unknowns by $\mathbf{x}$ and the right-hand side column by $\mathbf{b}$. The d linear equations can be used to express d unknowns as linear functions of the last unknown. In order to do this, a non-null minor of matrix $\mathbf{A_r}$ must be found.
Lemma 1 The linear system (6) has an infinite number of solutions when
$rank(\mathbf{A_r}) = rank(\mathbf{A_r}|\mathbf{b}) < d$,
has zero solutions when
$rank(\mathbf{A_r}) < rank(\mathbf{A_r}|\mathbf{b}) \leq d$,
can have up to two solutions when
$rank(\mathbf{A_r}) = rank(\mathbf{A_r}|\mathbf{b}) = d$.
Proof. Omitted.

For illustration of Lemma 1 see Fig. 2. Thus, if we find d minor of matrix $\mathbf{A_r}$ that is not zero, we can proceed in finding the coordinates of the center of the inscribed sphere.
We proceed by applying the ESSA algorithm to find a non-zero minor of $\mathbf{A_r}$ exactly. Without the loss of generality, assume that the minor of first d columns of $\mathbf{A_r}$ is non-zero. Denote this minor by $\mathbf{A}$. Thus, we can select the radius of the inscribed sphere as the free variable. Moving it to the right-hand-side, we get a linear system for d variables, representing the coordinates of the inscribed sphere. Applying Cramer's rule, we get the formulas for the coordinates:

$$\xi_i^* = \frac{A_i}{A}, i = 1..d, \tag{7}$$

where

$$A = \begin{vmatrix} x_{11}^* & x_{12}^* & ... & x_{1d}^* \\ x_{21}^* & x_{22}^* & ... & x_{2d}^* \\ ... & ... & ... & ... \\ x_{d1}^* & x_{d2}^* & ... & x_{dd}^* \end{vmatrix} \tag{8}$$

and A_i is obtained by replacing the i-th column of A by the column $\left[w_j^* - r_j^*\rho^*\right]_{j=1..d}$. Now we can substitute algebraic expressions for d unknowns xi_i^* into the last quadratic equation of the system (6). Multiplying the result by A^2, we arrive to:

$$A_1^2 + A_2^2 + ... + A_d^2 - A^2 = 0. \tag{9}$$

Note that each A_i depends on ρ^*. Now, we apply analytical transformations to equation (9) to transform it to a standard from of quadratic equation with coefficients a, b and c and unknown variable ρ^*. We must make sure that each of the long formulas is a sum of products of input variables, i.e. there are no fractions. Also, we need to return to original coordinates, obtaining the final solution is the form: $a\rho^2 + b\rho + c = 0$. The explicit formulas for the coefficients a, b and c are so complicated that they could not be obtained by Mathematica package. Thus, we implemented a subroutine to transform expressions analytically. Each of the coefficients, computed in the analytical form, contained approximately $3,000$ terms for 3D case. These expressions are omitted in the present discussion.

5 The Algorithm for Euclidean Voronoi Vertex Computation

This section presents the algorithm for the computation of the coordinates of the Voronoi diagram vertices, based on the obtained algebraic formulas. The algorithm proceeds by finding the roots of the resulting quadratic equation in exact floating-point arithmetic. The original

method of iterative approximation is based on the division of the interval, which can include the quadratic root, until the desired precision is reached. This, in turn, reduces to performing series of tests, computed exactly using the ESSA algorithm. The interval arithmetic is employed as needed.
In order to apply the ESSA to compute the sign of algebraic expression containing sums and products, the following approach is implemented. Consider the product of two single-precision numbers N_1 and N_2. The result is a monomial in double precision. It can be expressed as the sum of two single precision terms T_1 and T_2:

$$N_1 * N_2 = T_1 + T_2 \tag{10}$$

A product of three single-precision terms can be represented as four single-precision terms M_i, $i = 1..4$:

$$N_1 * N_2 * N_3 = (T_1 + T_2) * N_3 = \sum_{i=1}^{4} M_i \tag{11}$$

Similarly, a product of k single-precision terms $N_1 * N_2 * ... * N_k$ is represented as a sum of 2^{k-1} single precision monomials. Thus, the ESSA can be applied to compute the sign of algebraic expression containing sums and products exactly.

1. (Preprocessing) Determine the non-zero minor of matrix $\mathbf{A_r}$, by applying the interval arithmetic and ESSA (as needed). Obtain quadratic equation in the form $a\rho^2 + b\rho + c = 0$.

2. (Iteration) Determine with the precision to the last representable digit the root(s) of the quadratic equation.

   (a) Apply interval arithmetic to compute the roots ρ_1, ρ_2 of the quadratic equation. Denote the resulting intervals as $[a_1, a_2]$ and $[b_1, b_2]$ correspondingly. Three cases are possible:

      i. If interval $[d_1, d_2]$ representing the discriminant $d = b^2 - 4ac$ of the quadratic equation (6) is negative, i.e. $d_2 < 0$ - STOP: Report that no Voronoi vertex exists.

      ii. If $d_1 > 0$ GOTO Step 2.b

      iii. If $d_1 < 0$ and $d_2 > 0$, apply ESSA to compute the sign of d.
      - If $sign(d) < 0$ GOTO Step 2.a.i.
      - If $sign(d) > 0$ assign $d_1 = 0$, $d_2 = d_2$ GOTO Step 2.b
      - If $sign(d) = 0$ then double root $x_{12} = \frac{-b}{2a}$ exists. Replace quadratic equation with the linear equation in the form $b + 2ax = 0$. GOTO Step 2.b

   (b) Compute $sign(y(a_1))$, $sign(y(a_2))$, $sign(y(b_1))$, $sign(y(b_2))$ exactly using ESSA.

      i. If $sign(y(a_1)) \neq sign(y(a_2))$ GOTO Step 2.c (iterative computation of the root)

      ii. If $sign(y(b_1)) \neq sign(y(b_2))$ GOTO Step 2.c

      iii. If $sign(y(a_1)) = sign(y(a_2))$ (interval filter applied at Step 2.a was too rough), perform a linear search within the interval $[a_1, a_2]$ in half, until new values a_i, a_j are found satisfying condition $sign(y(a_i)) \neq sign(y(a_j))$.

      iv. If $sign(y(b_1)) = sign(y(b_2))$ - proceed in the same manner as above.

   (c) Apply ESSA iteratively to find root(s) exactly.

      i. Compute $mid = \frac{a_1 + a_2}{2}$ as a floating-point number.

      ii. If $mid = a_1$ or $mid = a_2$ STOP: Report mid as the solution.

      iii. Else compute $sign(mid)$ using ESSA, if $sign(mid) = 0$ STOP: Report mid as the solution.

      iv. Else if $sign(mid) \neq sign(a_2)$ assign $a_1 = mid$ GOTO 2.c; if $sign(mid) \neq sign(a_1)$ assign $a_2 = mid$ GOTO 2.c.

      v. Apply the same procedure to $[b_1, b_2]$.

Note. If the free variable determined by the preprocessing step is the radius of the inscribed sphere ρ, then an additional check to make sure that the radius is positive should be performed after Step 2.a. The situation when $[a_1, a_2] < 0$ or $[b_1, b_2] < 0$ corresponds to the existence of the outscribed sphere, which should be excluded from the consideration.
Lemma 2 The algorithm for reliable Euclidean Voronoi vertex computation will determine the radius of the inscribed sphere in d-dimensional space exactly with the precision to the last representable digit.
Proof. Omitted.

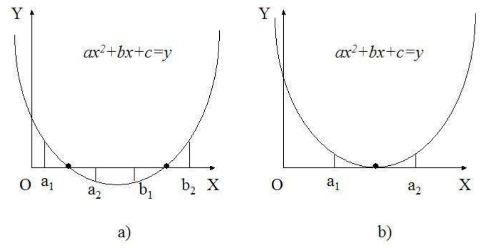

Figure 3: a) Two solutions of the quadratic equation; b) One solution of the quadratic equation

Finally, after the solution of the quadratic equation is obtained in the floating-point arithmetic, its value is substituted into linear expressions (7) in order to compute the remaining unknowns. Once again, interval filter applied first, followed by the ESSA.

Note that we reduced the problem of computing the intersection of hyperboloids exactly in the d-dimensional space, to finding the roots of a quadratic equation.

6 Conclusions

The algorithm is implemented for the 3-dimensional Voronoi diagram of a set of spheres. Experimentation is currently conducted for both random and degenerate configurations of spheres. Such parameters as the overall running time, the number of times ESSA is called, frequency of application of interval filter, the number of times when no solution or multiple solutions were found, etc. are considered. Similar statistics are also measured for a direct implementation of the method to compute the Voronoi vertex in Euclidean 3D space in imprecise floating-point arithmetic. Preliminary results obtained confirm the algorithm correctness and demonstrate suitability of the method for reliable computation of the Euclidean Voronoi diagram. The algorithm can be directly applied for reliable computation of other generalized Voronoi diagrams, including the Voronoi diagram in Laguerre geometry, and in L_1 and L_{inf} metrics.

7 Acknowledgements

The author would like to thank the University of Calgary Research Services for partial support of the project.

References

[1] Blum, L., Cucker, F., Shub, M. and Smale, S. "Complexity and Real Computation" (1997)

[2] Dey, T.K., Sugihara K. and Bajaj, C. L. DT in three dimensions with finite precision arithmetic, Comp. Aid. Geom. Des **9**(1992) 457-470

[3] Edelsbrunner, H. And Mcke, E. Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms, 4th Annual ACM Symposium on Computational Geometry (1988) 118-133.

[4] Fortune, S. And Wyk, C. Efficient exact arithmetic for computational geometry, 4th Annual ACM Symp. Comput. Geometry, (1993) 163-172

[5] Gavrilova, M. and Rokne, J. Reliable line segment intersection testing, Computer Aided Design, (2000) 32, 737-745.

[6] Gavrilova, M., Ratschek, H. and Rokne, J. Exact computation of Voronoi diagram and Delaunay triangulation, Reliable Computing, (2000) 6(1) 39-60.

[7] S. Krishnan, M. Foskey, T. Culver, J. Keyser, D. Manocha, PRECISE: Efficient Multiprecision Evaluation of Algebraic Roots and Predicates for Reliable Geometric Computations, Symp. on Computat. Geometry (2002)

[8] Naher, S. The LEDA user manual, Version 3.1 (Jan. 16, 1995). Available from ftp.mpi-sb.mpg.de in directory /pub/LEDA.

[9] Ratschek, H. And Rokne, J. Exact computation of the sign of a finite sum, Applied Mathematics and Computation, (1999) 99, 99-127.

[10] Rokne, J.: *Interval arithmetic.* In: Graphics Gems III, Academic Press, pp. 61-66 and pp. 454-457 (1992).

[11] Sugihara, K. And Iri, M. A robust topology-oriented incremental algorithm for Voronoi diagrams, IJCGA, (1994) 4 (2): 179-228.

[12] Yap, C., Dub, T. The exact computation paradigm, In "Computing in Euclidean Geometry" (2nd Edition). Eds. D.-Z. Du and F.K. Hwang, World Scientific Press (1995)