Title:

Voronoi Cells of Non-general Position Spheres using the GPU

Authors:
Zhongyin Hu, daphnedut@gmail.com, University of California, Berkeley
Xiang Li, xli@berkeley.edu, University of California, Berkeley
Adarsh Krishnamurthy, adarsh@iastate.edu, Iowa State University
Iddo Hanniel, iddohanniel@gmail.com, Technion, Israel Institute of Technology
Sara McMains, mcmains@me.berkeley.edu, University of California, Berkeley

Keywords:
Voronoi diagram, spheres, GPU, lower envelope

DOI: 10.14733/cadconfP.2016.173-177

Introduction:

The Voronoi diagram is a fundamental construct in computational geometry, for which many algorithms have been proposed. However, Voronoi diagrams of large collections of higher order objects such as spheres in R^3 have only been successfully computed by Kim et al. [3], which is limited to input in general position. Our main contributions include:

1. A novel approach to compute Voronoi cells of spheres that exploits the parallelism of the GPU by shooting intersection rays and taking the lower envelope as points on Voronoi faces.
2. Separation of the construction of the topology of the Voronoi cells from the calculation of the geometry. The topology is calculated directly from the face sample points.
3. Accurate calculation of Voronoi vertices' geometry by using the samples to initialize Newton-Raphson iteration.
4. Calculation of Voronoi cells of thousands of input spheres representing actual protein molecules can be generated by our algorithm. Our algorithm is robust even for spheres not in general position, handling Voronoi vertices with degree greater than four.

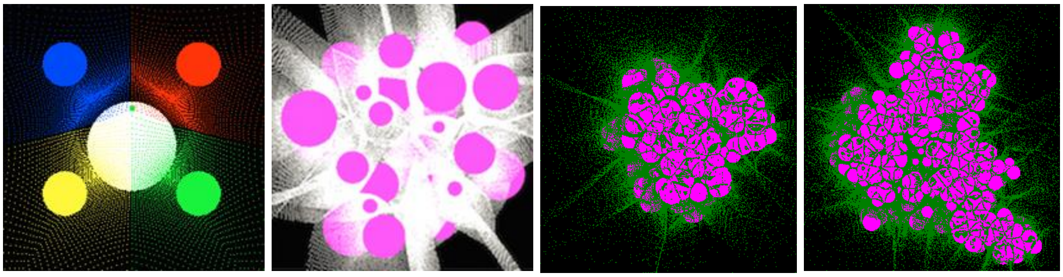

Fig. 1: Example results: (a) A single Voronoi cell pictured, where the green Voronoi vertex is equidistant from the 5 spheres; the full Voronoi diagram we computed for (b) "Random Set A" [4]; (c) Protein ID 1crn-PQR consisting of 642 atoms [5]; (d) Protein ID 1bh8-PQR consisting of 2161 atoms [5].

Algorithm Description:

Definition 1: Given a set of spheres S_0, S_1,…, S_n in R^3, the Voronoi cell of sphere S_i is the set of all points closer to S_i than to S_j ($\forall j \neq i$). The Voronoi diagram is the union of the Voronoi cells of all (n + 1) spheres.

Definition 2: The locus of points that are equidistant from S_i and S_j ($j \neq i$) is called bisector $B_{i,j}$. In general, bisectors in R^3 are surfaces, either a plane when the two spheres are the same size, or a hyperbolic surface when the two spheres are different sizes [2].

We assume that no input sphere is completely contained inside another, but our same framework handles intersecting spheres correctly. Voronoi faces are portions of such bisector surfaces between two spheres that have equal (minimal) distance from the two spheres. Please refer to Elber et al. [1] for a detailed formulation of the bisectors.

In this paper, instead of analytically calculating the Voronoi faces, we propose a sample-based approach to obtain sample points on Voronoi faces, by taking the lower envelope of the bisectors with respect to the distance function to the "base sphere" in each Voronoi cell. We shoot sample rays from this sphere in radial projection directions, calculate the intersection of each ray with the implicit bisector surface functions, and retain only the intersection with the minimum distance as our face sample point.

We calculate the Voronoi vertices of each cell by using a marching-grid approach to locate the neighborhoods of the vertices. We virtually "color code" each Voronoi face sample point by giving it the same color as the bisector to which it belongs. A Voronoi vertex occurs where there are three or four different colors out of four neighboring sample points, which means there is an intersection of three or more bisectors.

The stages of the algorithm to construct a Voronoi cell are:

1. Sample rays from the base sphere S_i.
2. Calculate the bisector surfaces functions between S_i and S_j ($j = 0, 1, \ldots, i-1, i+1, \ldots, n$).
3. Compute the intersection of each ray with all the bisectors and take the lower envelope of all the intersections to obtain the sample points on the Voronoi faces.
4. Find all grid cells of neighboring sample points that contain Voronoi vertices and use the Newton-Raphson method to calculate the vertex location.

Sampling Rays from Base Spheres

For simplicity of calculation, we transform the coordinate system so that each base sphere in turn is a unit sphere with radius 1 located at the origin. We sample normal rays from the sphere surface (Fig. 2(a)) using a parametric representation of the ray:

$$\mathbf{r}(t) = \boldsymbol{r}\big(x(t), y(t), z(t)\big) = \boldsymbol{O} + t \cdot \boldsymbol{n} \tag{1}$$

For sampling, we parameterize the six faces of the axis-aligned bounding cube of the base sphere. Each face of the bounding cube is taken as a uniformly subdivided parameterized domain expressed in variables u and v (Fig. 2(b)).

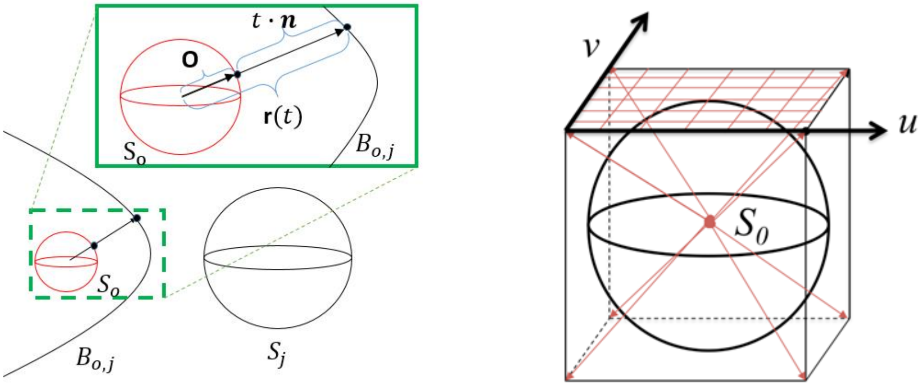

Fig. 2: (a) Illustration of ray intersection with bisector surfaces. (b) Mapping a base sphere to six u-v parametric surfaces on the bounding cube; uniform parametric sampling of top surface shown.

Calculating the Bisector Functions

The bisector surfaces between two spheres are the simultaneous solutions of two distance equations; the solution can be expressed as the implicit quadratic surface:

$$Ax^2 + By^2 + Cz^2 + Dxy + Exz + Fyz + Gx + Hy + Iz + J = 0 \qquad (2)$$

where $A = 4R^2 - 4D_x^2$, $B = 4R^2 - 4D_y^2$, $C = 4R^2 - 4D_z^2$, $D = -8D_xD_y$, $E = -8D_yD_z$, $F = -8D_zD_x$, $G = -8R^2C_{x1} - 4KD_x$, $H = -8R^2C_{y1} - 4KD_y$, $I = -8R^2C_{z1} - 4KD_z$, and $J = 4R^2(C_{x1}^2 + C_{y1}^2 + C_{z1}^2) - K^2$, where $R = R_1 - R_2$, the difference in the radii, $D_x = C_{x1} - C_{x2}$, $D_y = C_{y1} - C_{y2}$, $D_z = C_{z1}- C_{z2}$, the distance between the centers, and $K = (C_{x2}^2 - C_{x1}^2) + (C_{y2}^2 - C_{y1}^2) + (C_{z2}^2 - C_{z1}^2) - R^2$. If the two spheres are different sizes, the bisector is a hyperbolic surface; otherwise it is a plane.

Finding Intersections and Lower Envelope

The Euclidean distance function r of a point (x, y, z) from a sphere surface is $(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = (r + r_0)^2$, where (x_0, y_0, z_0) is the center of the sphere with radius r_0. When solving for the intersection of the ray and the bisector surface, we substitute the ray representation (Eq. (1)) in terms of t into the bisector implicit function (Eq. (2)); which can be simplified as a quadratic equation in parameter t when the bisector is a hyperbolic surface (i.e. $a \cdot t^2 + b \cdot t + c = 0$), or a linear equation when the bisector surface is a plane (i.e. $d \cdot t + e = 0$). The solution to the hyperbolic bisector equation has two possible sheets, and depending on the relative radius of the second sphere to the base sphere, only the solution closer to the smaller sphere should be retained (see Fig. 3(a)).

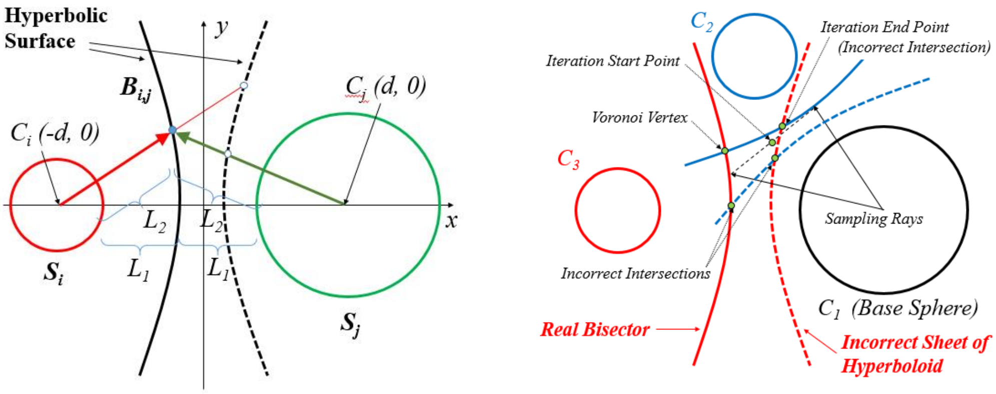

Fig. 3: (a) Culling the incorrect sheet of the two hyperboloid sheets. (b) Converging to an incorrect intersection with an incorrect sheet of the bisectors.

Calculating Voronoi Vertices:

We use a marching-grid approach to locate the neighborhood of Voronoi vertices by checking each group of four neighboring face sample points, which we call a “grid cell” on the bounding cube. In the u-v parametric domain, for each such grid cell, we color-code each of its four corner points based on the corresponding bisector. Within a grid cell, if the four corners have three or more colors, then they correspond to three or more bisectors and thus to a corresponding number of Voronoi faces. Hence, at least one Voronoi vertex must be located within the cell (e.g. where those Voronoi faces intersect).

Fig. 4(a) and 4(b) show the correspondence between sample points on Voronoi faces in geometric space and the samples on the bounding cube of the base sphere in the u-v domain. The Voronoi cell of the base sphere (the large sphere in the middle) is shown on the left; it has four surrounding unbounded Voronoi faces (red, blue, green, and yellow). Infinite rays are represented by the gray portion in the color map of the bounding cube (Fig. 4(b)). Fig. 4(c) shows one of the u-v domains of the bounding cube of the base sphere. The grid cells indicated contain three colors.

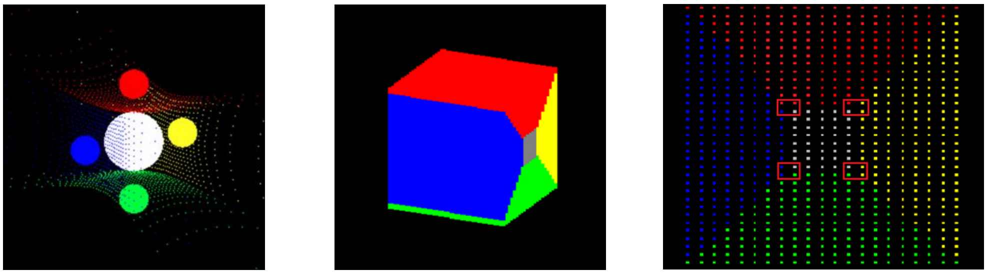

Fig. 4: (a) Sample points on Voronoi faces of the Voronoi cell of the white base sphere; (b) corresponding color map of u-v domains on bounding cube; (c) sample point grid on one face of the bounding cube with 3-color grid cells indicated.

For a three-color grid cell, finding the Voronoi vertex is equivalent to solving the three bisector surface equations simultaneously. We use the Newton-Raphson method to find locations of the actual vertices. For the start point for the Newton-Raphson method, we intersect a ray that is the average of the four corner points of the grid cell with the corresponding bisector surfaces. Occasionally this leads to converging to a solution at the intersection of an incorrect sheet of a hyperbolic surface (Fig. 3(b)). We identify such cases by confirming that the solution is truly equidistant to the spheres defining the corresponding bisectors, and if not, reinitialize with the ray intersection with a different bisector. In practice we found that the iteration converges within 10 steps except in cases such as shown in Fig. 5, where the iteration oscillates between two intersections of the same bisector surfaces close together. When we identify such non-convergence, we subdivide the grid cell.

We uniformly subdivide the original grid cell into four sub grid cells. Taking the lower envelope of each new ray with bisectors gives us five new sample points. We identify if any of the sub-cells have the same three colors as their parent. If so, we take the average of the rays of such cells as a new start point for the iteration. Otherwise, when all of the four children share only one or two colors with the parent, there is no vertex to output.

When there is a new color identified during the subdivision, there is a new Voronoi face that was missed in the original sampling. If the new color lies on an edge of the old grid cell, we also subdivide the neighboring grid cell sharing the edge. Subdivision is also required to distinguish the special cases where all four corners of the grid cell are different colors. There are two possibilities: 1) the four corresponding bisector surfaces intersect at one vertex; or 2) there is more than one Voronoi vertex in the neighborhood and we then increase the sampling density for this grid cell to identify the other Voronoi vertices.

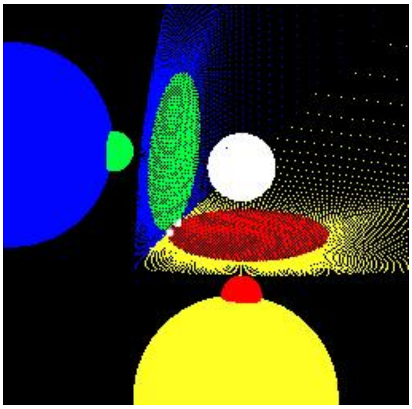

Fig. 5: Case of a non-converging grid cell with two intersections: the two Voronoi vertices (white) are at the intersection of the same four bisectors (blue, green, red, and yellow).

Results:

Our algorithm to compute the Voronoi cells was run on a PC with an Intel(R) Core(TM) 2 Quad CPU Q9400 @2.66GHz with 4.0 GB RAM and an NVIDIA Quadro 6000 graphics card.

Our algorithm is robust in handling special cases of Voronoi vertices that arise with spheres not in general position, such as those that arise with symmetrical input, as in the cell in Fig. 1(a) with a degree-4 vertex. In addition to such synthetic test cases (e.g. Fig. 1(a) and 1(b)), we tested our implementation with several examples of each of two types of protein structures from the protein data bank [5]: PDB format and PQR format. In PDB format, atom spheres usually have different radii even for the same element, whereas the PQR format includes hydrogen atoms that all have the same radii. The hydrogen atoms greatly increase the presence of local symmetrical patterns, which can lead to non-general position Voronoi vertices.

To test our ability to effectively build Voronoi diagrams for inputs of hundreds or thousands of spheres and handle Voronoi vertices not in general position, we selected protein "1JD0" under PDB format, protein "1al1" and "1bh8" under PQR format, and protein "1crn" under both formats. Tab. 1 shows the total running time for computing the Voronoi cells for these proteins as sampling density increases. Running times increase roughly linearly with the number of input spheres (atoms), and sublinearly with the number of sample rays, which are processed in parallel on the GPU

Sampling	*1al1-PQR (217 atoms)*	*1crn-PDB (327 atoms)*	*1crn-PQR (642 atoms)*	*1bh8-PQR (2161 atoms)*	*1JD0-PDB (4195 atoms)*
30*30	1.30	2.02	3.85	23.3	50.2
60*60	1.92	3.06	5.83	35.4	81.4
80*80	2.29	3.85	7.05	41.2	93.6
100*100	2.45	4.21	7.82	45.9	102

Tab. 1: Total computation time under different sampling rates. Running time in seconds.

Conclusion:

We have presented a new approach to calculating Voronoi cells of spheres that combines a sample-based, lower-envelope approach with numerical iteration. The numerical iteration allows us to calculate the geometry of the vertices to a much greater accuracy than the ray sampling density. The lower-envelope calculations for sample rays are able to exploit the parallelism of the GPU, and are robust for non-general position input.

References:

[1] Elber, G.; Kim, M.-S: Computing rational bisectors, Computer Graphics and Applications, IEEE 19(6), 1999, 76–81. http://dx.doi.org/10.1109/38.799747

[2] Hanniel, I.; Elber, G.: Computing the Voronoi cells of planes, spheres and cylinders in R^3, Computer Aided Geometric Design, 26(6), 2009, 695–710. http://dx.doi.org/10.1016/j.cagd.2008.09.010

[3] Kim, D.-S; Cho, Y.; Kim, D: Euclidean Voronoi diagram of 3D balls and its computation via tracing edges, Computer-Aided Design, 37(13), 2005 1412–1424. http://dx.doi.org/10.1016/j.cad.2005.02.013

[4] Random spheres data set from Voronoi Diagram Research Center, http://voronoi.hanyang.ac.kr/qtdb/testdataset/random_sphereset.htm, accessed: 2015-04-05.

[5] The RCSB Protein Data Bank, http://www.rcsb.org/pdb/home/home.do, accessed: 2016-02-25.