ON A NEAREST-NEIGHBOR PROBLEM IN MINKOWSKI AND POWER METRICS

MARINA GAVRILOVA

Department of Computer Science, University of Calgary,
Calgary, AB, Canada, T2N-1N4
e-mail: marina@cpsc.ucalgary.ca

Abstract: This report presents an efficient algorithm based on the generalized Voronoi diagram construction for solving the nearest-neighbor problem in the plane. The input for the problem is the set of circular sites S with varied radii, the query point p and the metric (Minkowski or power) according to which the site, neighboring the query point, is to be reported. The IDG/NNM software was created that allowed the experimental study of the posed problem. The performance of the method was tested on large data sets with such parameters as the number of sites, distribution of their radii, site configuration, density of packing, and the number of searches, varied. The efficiency of the generalized Voronoi diagram method was compared against the k-d tree method for finding the nearest-neighbor. The results demonstrate that the Voronoi diagram method outperforms the k-d tree method for all testes input site configurations, including the silo granular-type material model. Finally, the similarity between the nearest-neighbor relationship in the Minkowski and power metrics is worth noticing.

Keywords: nearest-neighbor queries, generalized Voronoi diagrams, Minkowski and power metrics, Laguerre geometry, k-d trees, granular-material simulation.

1. Introduction

The Voronoi diagram is often used as a convenient tool for solving scientific problems in computer modeling of physical phenomena. These include structure analysis of unordered systems (liquids, solutions, polymers) [17], stress analysis and simulation of granular systems (ice flow, silo models) [9], and space structure in complex molecular and biological systems [13].

There are some challenges arising while investigating such problems. The existing software and algorithms are not customized to efficiently solve a variety of application problems. A particular problem addressed in this report is finding a nearest-neighbor in the system of poly-sized objects of the circular shape [11]. The application of the ordinary point site Voronoi diagram to perform the nearest-neighbor query in 2D is straightforward [1]. The algorithm takes $O(n)$ space and the worst-case running time of $O(n)$. The same idea has been extended to higher dimensions [14], applied to solve the point location problem among the convex sites [15], and used to solve the nearest-neighbor problem for dynamic Voronoi diagram [12]. The generalized VD in Laguerre geometry was successfully used to solve the collision optimization problem in the system of moving particles [7]. The properties of the generalized weighted Voronoi diagrams that allow the use of this data structure for nearest-neighbor detection were thoroughly investigated in [6]. However, there hasn't been a study that compares the various generalized Voronoi diagrams on applicability for solving the nearest-neighbor problem.

Thus, this report presents a study of the generalized Voronoi diagram approach for finding the nearest-neighbor for a set of non-intersecting spheres. Application domains for this problem can be found in computer graphics, GIS, computer modeling and computer simulation [11]. The developed method can also be applied to problems from statistics and information retrieval fields. The data structures studied are generalized weighted Voronoi diagram under the Manhattan, supremum and power metrics. The main result is the efficient and robust algorithm for nearest-neighbor computation in Manhattan, supremum and power metrics that is successfully applied to the silo model simulation in a framework of the granular-type materials model.

The performance of the VD based method was compared against the CORE library implementation of the k-d tree method [2], modified to handle the weighted sites under the supremum metrics. The test results show the significantly better performance of the generalized VD based method on various test data, including large (10,000 sites) data sets with various topologies. It was also established that in 90% to 95% of considered cases the same nearest neighbor is reported by VD method in different metrics. Finally, it's worth noting that the similar technique also allows solving a variety of problems such as all nearest-neighbors, point location and range queries.

2. Defining the problem

Consider a set of circular sites S in the plane. Define a *nearest-neighbor* relation between the query point *x* and a site *P* as follows. The point $x \in R^2$ is the nearest neighbor of $P \in S$ iff $d(x,P) \le \min_{Q \neq P} d(x,Q)$, $Q \in S$ [8]. The distance $d(x,P)$ between a point $x(x_1, x_2)$ and a sphere $P = \{p, r_p\}$ with the center at $p(p_1, p_2)$ and radius r_p can be computed as

$$d(x,P) = d(x,p) - r_p = \sum_{i=1}^{2} |x_i - p_i|$$

in the Manhattan (L_1) metric and as

$$d(x,p) = \max_{i=1..d} |x_i - p_i|$$

in the *supremum* (L_∞) metric. It is computed according to the formula:

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

in the Laguerre geometry (under the *power* distance function).

As a tool to solve the problem we propose to use the generalized Voronoi diagram. The *generalized Voronoi diagram* for a set of circles *S* in the plane is a set of generalized Voronoi regions, where $GVor(P) = \{x | d(x,P) \le d(x,Q), \forall Q \in S - \{P\}\}$ and $d(x,P)$ is the distance function between a point x and a circle *P* [11]. The distance function $d(x,P)$ is defined according to the metric employed (see fig. 1).

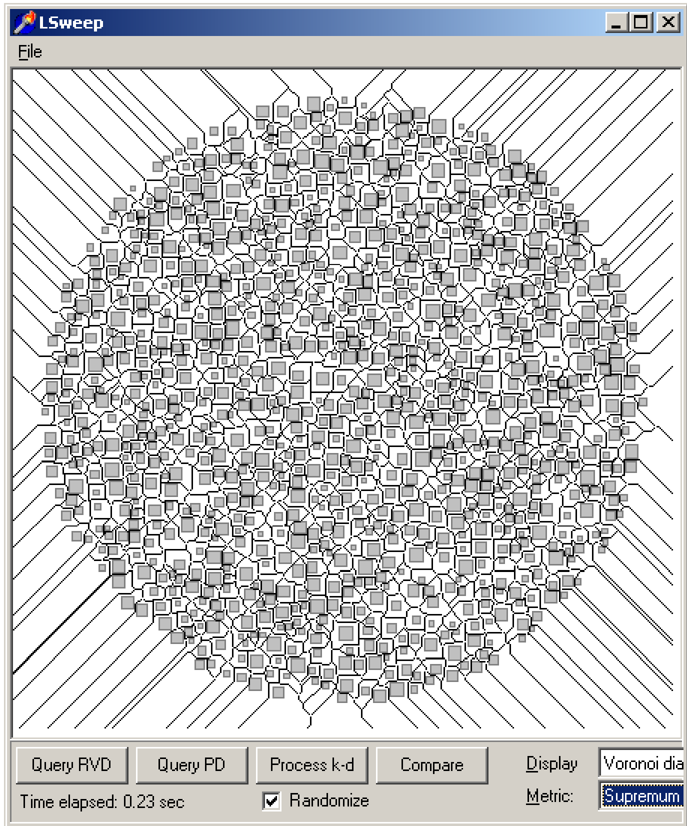


Figure 1. Supremum weighted VD for 1000 randomly distributed sites generated using Initial Distribution Generator (IDG) module.

A *generalized Delaunay triangulation* is the dual of the generalized Voronoi diagram obtained by joining all pairs of sites whose generalized Voronoi regions share a common generalized Voronoi edge.

The nearest-neighbor property for VD in various metrics, including Manhattan, supremum and power, was established in [7]. This property allows solving the nearest-neighbor problem by constructing the generalized Voronoi diagram or the Delaunay triangulation. The method is presented in the following section.

3. The nearest-neighbor search algorithm

The following outlines the nearest-neighbor search algorithm based on the generalized VD construction. The input for the algorithm is a set of circular sites. The approach is based on the simple edge-walk technique that starts with the random location in the Voronoi diagram:

Step 1 (initialization) Build the weighted generalized Delaunay tessellation using a flip based incremental construction technique.

Step 2 Find a site *P* neighboring the query point *x*.

2.1. Randomly select a site P as a starting site for the search (call it an *active* site). Initially set a random edge of the DT as the value for three variable $e_{prev}=e_{curr}=e_{new}$.

2.2. Randomly select an edge adjacent to the *active* site in DT.

2.3. Perform a counter-clock wise walk alone the DT edges adjacent to the active site. Select the first edge such that $\boldsymbol{x}$ is located to the left of the straight line passing through this edge, by performing the CCW orientation test.

2.4. Update $e_{prev} = e_{curr}$, $e_{curr} = e_{new}$. Set e_{new} as the newly found edge.

2.5 If edges e_{prev}, e_{curr}, e_{new} do not form a triangle enclosing the query point $\boldsymbol{x}$, set an endpoint of e_{new} edge to be a new *active* site, GOTO step 2.2.

Step 3. Report the closest of the vertices of the triangle formed by the edges e_{prev}, e_{curr}, e_{new} as the nearest-neighbor to the query point x.

The preprocessing step is worst-case optimal $O(nlogn)$, the worst-case number of edges visited during the Step 2 is $O(n)$ (since we never consider the visited edge twice) and the space complexity is $O(n)$.

Note 1. The description of the incremental flipping algorithm can be found in [3].

Note 2. The presented algorithm is applicable for solving the point location problem and the range search problem in the presence of specific constraints. Thus, the presented algorithm locates the Voronoi region containing the query point and the generator of this Voronoi region is reported as the nearest neighbor.

Note 3. The method is extendable to higher dimensions, where a slight modification of the Step 2 is required to perform a point location with respect to the *d-1* hyperplane.

Note 4. The Voronoi diagram does not depends on the sizes or distributions of the spheres, with the exception of close to degenerate cases, that also require special treatment in the cell-based or *k-d* tree methods [2].

4. IDG/NNM software

The outlined above algorithm was implemented in the object-oriented Borland Delphi environment. The experiments were conducted on a Pentium II 350 computer with 128 MB RAM.

The program consists of two modules. The first module, the Initial Distribution Generator (IDG), is used to create the various configurations for the input sites. IDG allows generation of a new distribution by importing the distribution from a text file, where the coordinates of the centers and radii of circles are specified. IDG can also automatically generate various distributions, such as uniform distribution of sites in the interior of a square, uniform distribution of sites in the interior of a circle, cross, ring, corners of the square, degenerate grid and degenerate circle. Then, the parameters of the random distribution, such as the number of circles, the distribution of their radii, the size of the area, and the type of the distribution must be specified. The second module, the Nearest-Neighbor Monitor (NNM) is the program that constructs the additively weighted supremum VD, the power diagram and the *k-d* tree in supremum metric for the specified input file. Then NNM performs a series of nearest-neighbor searches of the specified length.

The efficiency of the VD base method was compared against the *k-d* tree method for a set of circles in the plane. The *k-d* tree implementation is based on the *Ranger* software [10], which implements an *optimal k-d tree method* for a set of points [4]. The software was modified to accommodate for the circular sites. Each circular site was modeled by four corner points, thus the modified k-d tree method reported the nearest neighbor in the weighted supremum metric. The software was also optimized to avoid the unnecessary memory allocations and initializations of the internal variables for maximum efficiency when performing multiple queries. The efficiency of the method does not depend on the metric used, thus the supremum and power VD methods were compared to the same implementation of the *k-d* tree method.

After the initial distribution is generated, the test file is loaded into the NNM Module. First, the generalized Voronoi diagram, Delaunay Triangulation or a *k-d* tree is computed in power or supremum metric. The snapshot of the screen presented in Fig. 4 illustrates the Voronoi diagram in Laguerre geometry constructed for 10000 circular sites in ring configuration with radii varied from 0.000000001 to 0.05 pts.

The nearest-neighbor queries are done by either generating a sequence of random queries or by performing on-line queries by selecting the query point manually. When an on-line query is performed, the path from the starting VD edge to the nearest-neighbor of the query point is highlighted and the length of this path is displayed. The measured characteristics, determining the performance of the method, include the total number of queries performed, the elapsed time, the time per one query, and the *average search length*. The average search length is a parameter related to the number of comparisons that algorithm performs. For the VD approach, this parameter represents the total number of edges that were encountered during an edge walk while performing a query. In case of *k-d* tree this parameter represents the total number of distance comparisons performed on different nodes of the tree during the query. This parameter was selected for evaluation since it not only allows to compare VD and *k-d* tree methods, but also can be easily visualized.

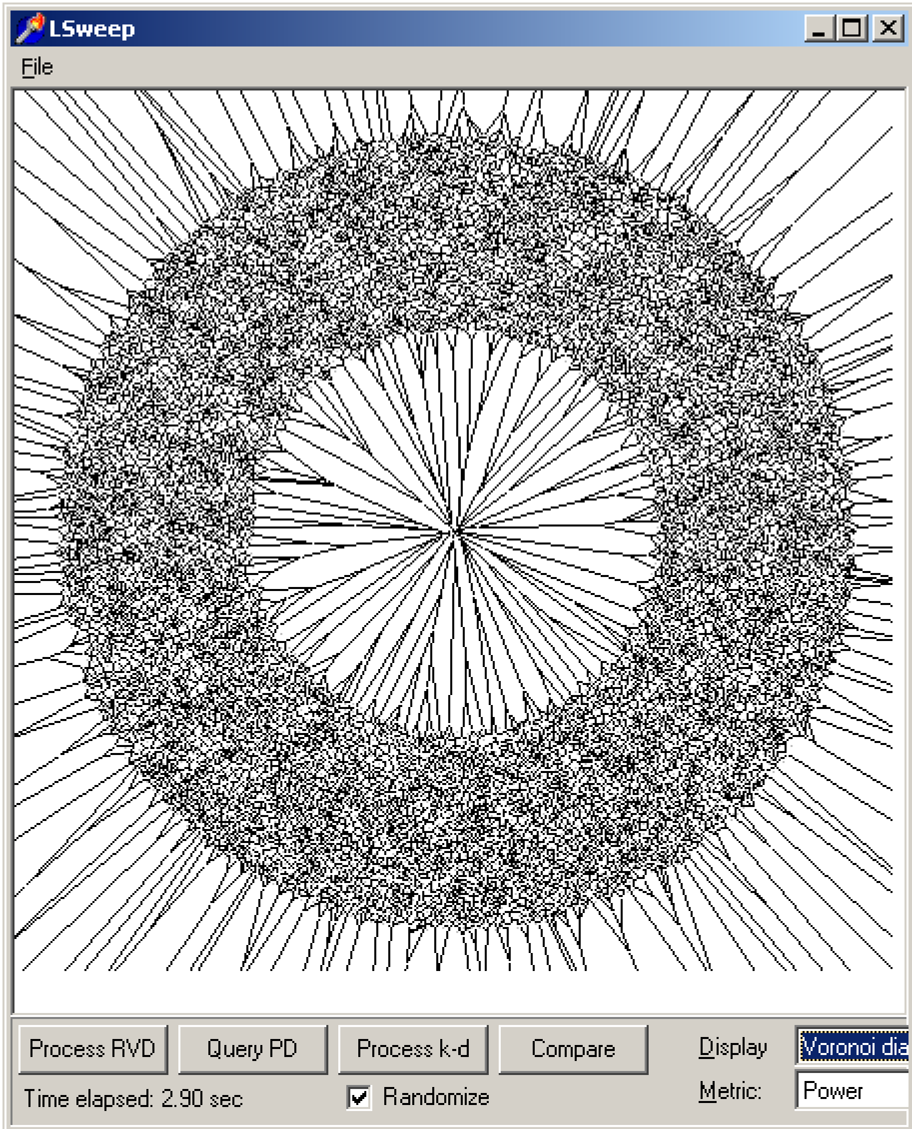


Figure 4. Example of generalized VD under power metric for 10000 sites, ring configuration.

5. Experimental studies of the nearest-neighbor search algorithm

The experiments were performed for different data set sizes, various distributions of their density, radii and site configurations. The algorithms were tested on the generated data sets and the data set representing granular-type material system for a silos simulation with large number of particles[1].

---

[1] Data was provided by the Dept. of Mechanical Engineering, University of Calgary.

The first series of experiments were performed on randomly generated distributions such as uniform square, uniform circle, cross, ring, degenerate grid and degenerate circle distributions. All the distributions were tested for 100 to 10000 input sites. The experiments show that the *k-d* tree method requires much less initialization time than the Voronoi diagram methods, even though the upper bound for both algorithms is *O(n)*.

The test results showed that both the power diagram and supremum diagram method consistently outperformed the *k-d* tree method in terms of the query time (see fig. 5 for specific configurations). Such behavior also could be related to the fact that most of the experiments were done on the close to degenerate configurations. Note that the running time is the closest for the degenerate circle distribution, when all the sites are located on the circle with an empty interior.

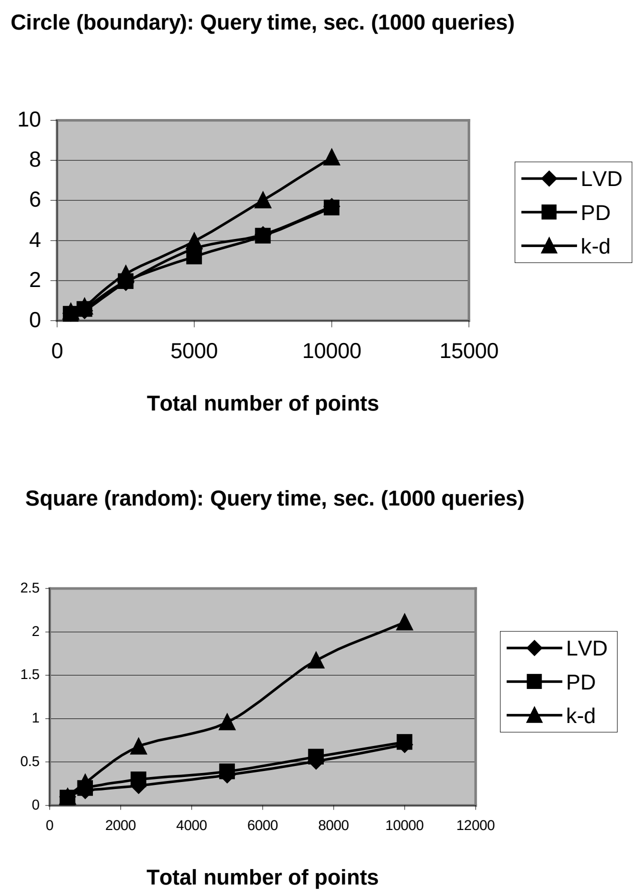


Figure 5. Time required for performing 1000 queries vs. the number of sites.

The average search length was recorded for all the tests performed and it showed the same linear dependence (growth) as the number of sites increased. Thus, for uniform and degenerate grid distributions it increases from 10 for 100 sites to about 180 for 10000 sites. In case of circle distribution it increases from 1 for 100 sites to 5500 for 10000 sites. This result explains that the queries on the circle distribution are usually more time consuming than queries performed on all others distributions.

Based on the results obtained, the conclusion can be made that the power and supremum Voronoi diagram method is an efficient data structure for performing nearest-neighbor queries, independent of the site configurations. This dependency does not change as the number of input sites increases from 100 to 10000. However, the initialization time for VD is quite large compared to the *k-d* tree method. Also, as it would be expected, more time is required for the initialization of the data structure and for performing the queries for degenerate distributions.

Another interesting result is that the VD in either metric can be used for approximate nearest-neighbor searches. The following series of experiments were performed to determine how 'close' the results reported by the power diagram method will be to the supremum method. The experimental results are summarized in the following table (table 1):

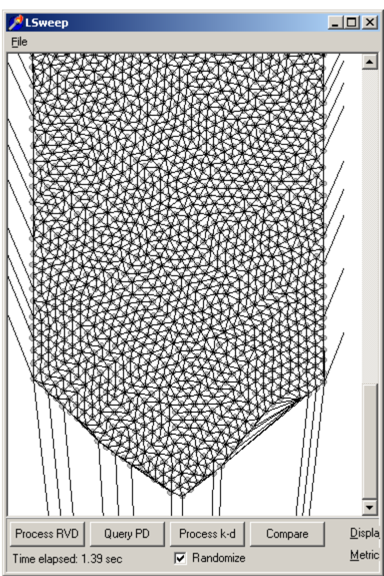
Neighbors reported	**Percentage**
0 (same result)	90-95%
1 (neighbors in DT)	5-10%
2	0.01-0.05%
3 or more	0% (never encountered in 50000 queries)

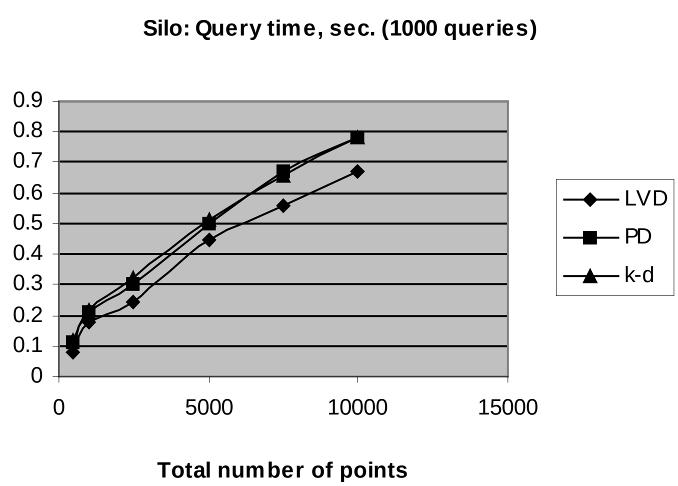
Table 1. Neighbors reported by supremum and power VD methods

This shows that it is possible to use either a power diagram or a supremum Voronoi diagram for *approximate* nearest-neighbor search.

The third series of the experiments were performed on a data set generated as a result of computer simulation of the granular-type material system for a silo model [5]. The model represents a grain elevator with vertical boundaries and a large number of densely packed grain particles. The test results show that the initialization time for the power diagram method is practically the same as for the *k-d* tree method and the supremum diagram requires significantly more time for initialization. The query time for power diagram is almost the same as for supremum diagram and significantly outperforms the *k-d* tree method (see fig. 6).

Figure 6. DT built in power metric for 2500 particles and the running time for 1000 queries.





6. Conclusion

This report presented an algorithm and its implementation for efficient solution of the nearest-neighbor problem for a set of weighted sites through the generalized Voronoi diagram. Obtained results clearly demonstrate the applicability of the generalized VD under various distance functions as an efficient, robust and easy to implement model for solving the nearest-neighbor problem.

7. References

1. Aggarwal and P. Raghawan, Deferred data structures for the nearest-neighbor problem, Inform. Process. Letters, vol. 40, num. 3, 1991, pp. 119-122.
2. Bentley, J. L. 1990. "K-d Trees for Semidynamic Point Sets," in Proceedings of the 6th Annual ACM Symposium on Computational Geometry, pp. 187-197.
3. Edelsbrunner, H. and Shah, N. 1996. "Incremental topological flipping works for regular triangulations," Algorithmica, vol. 15, pp. 223-241.
4. Friedman, J., Bentley, J. and Finkel, R. 1977. "An Algorithm for Finding Best Matches in Logarithmic Expected Time," ACM Transactions on Mathematical Software, vol 3. no. 3, pp. 209-226.
5. Gavrilov, D. and Vinogradov, O. 1997. "A cluster in granular systems as a topologically variable structure," in Proceedings of 1997 ASCE Symposium on Mechanics of Deformation and Flow of Particulate Materials, Evanston, IL, pp. 299-307.
6. Gavrilova, M. 1999. "Proximity and Applications in General Metrics," Ph. D. Thesis, Dept. of Computer Science, University of Calgary, Canada.
7. Gavrilova, M., Rokne, J., Vinogradov, O and Gavrilov, D. 1999. "Collision detection algorithms in simulation of granular materials," 1999 ASME Mechanics and Materials Conference, pp. 283- 284.
8. Joseph O'Rourke, 1994 Computational geometry in C. Cambridge University Press.
9. Luchnikov, V.A., Medvedev, N.N., Oger, L. and Troadec, J.-P. 1999. "The Voronoi-Delaunay analysis of voids in system of nonspherical particles." Phys.Rev.E., v.59(6), p.7205-7212.
10. Murphy, M. and Skiena, S. 1996. "A study of data structures for orthogonal range and nearest neighbor queries in high dimensional spaces," CSE 523/524 Master's Project, Department of Computer Science, SUNYSB.
11. Okabe, A., Boots, B. and Sugihara, K. 1992. Spatial Tessellations: Concepts and Applications of Voronoi Diagrams. John Wiley & Sons, Chichester, England.
12. O. Devillers, M. Golin, K. Kedem and S. Schirra "Queries on Voronoi Diagrams of Moving Points," Comput. Geom. Theory and Applications, vol. 6, 1996, pp. 315-327.
13. Paschek, D. and Geiger, A. 1999. "Simulation Study on the Diffusive Motion in Deeply Supercooled Water." J.Phys.Chem. B., v.103(20), 4139-4146.
14. S. Berchtold, B. Ertl, D. Keirn, H.-P. Kriegel, T. Seidel "Fast nearest neighbor search in high-dimensional space", in Proc. Of the 14th Intn. Conf. On Data Emg, Orlando, Florida, 1998.
15. Schaudt, B. and Drysdale, R. 1992. "Higher-dimensional Voronoi diagrams for convex distance functions," in Proceedings of the 4th Canadian Conference on Computational Geometry, pp. 274-279.
16. T.Graf and K. Hinrichs, A Plane-Sweep Algorithm for the All-Nearest-Neighbors Problem for a Set of Convex Planar Objects, in Proc. 3rd Workshop Algorithms Data Structures, LNCS, vol. 709, Springer-Verlag, 1993, pp. 349-360.
17. Voloshin, V.P. and Medvedev, N.N., 2000. "Void space analysis of the structure of liquids." Annual Conference on the Physical Chemistry of Liquids: Molecules - Macromolecules - Biomolecules. September 8-13. Regensburg, Germany.