International Journal of CAD/CAM Vol. 5, No. 1, pp. 59 ~ 68 (2005)

INTERNATIONAL JOURNAL OF CAD/CAM
www.ijcc.org

Topology Representation for the Voronoi Diagram of 3D Spheres

Youngsong Cho, Donguk Kim and Deok-Soo Kim*[1]
Voronoi Diagram Research Center, Hanyang University, Seoul, Korea
[1]Department of Industrial Engineering, Hanyang University, Seoul, Korea

Abstract – Euclidean Voronoi diagram of spheres in 3-dimensional space has not been explored as much as it deserves even though it has significant potential impacts on diverse applications in both science and engineering. In addition, studies on the data structure for its topology have not been reported yet. Presented in this paper is the topological representation for Euclidean Voronoi diagram of spheres which is a typical non-manifold model. The proposed representation is a variation of radial edge data structure capable of dealing with the topological characteristics of Euclidean Voronoi diagram of spheres distinguished from those of a general non-manifold model and Euclidean Voronoi diagram of points. Various topological queries for the spatial reasoning on the representation are also presented as a sequence of adjacency relationships among topological entities. The time and storage complexities of the proposed representation are analyzed.

Keywords: **Euclidean Voronoi diagram of spheres, Non-manifold model, Data structure**

1. Introduction

The Voronoi diagram has been one of the central topics in computational geometry and CAD/CAM due to its diverse applications in various scientific and engineering disciplines. An ordinary Voronoi diagram for a point set has been extensively studied and its properties are well-known in 2 and higher dimensions [27]. However, Voronoi diagrams for circles in 2D and spheres in 3 or higher dimensions in Euclidean distance measure have not been explored as much as they deserve even though they have significant potential impacts on diverse applications in both science and engineering [1, 10, 15, 16, 25, 26, 30, 32].

An efficient and robust algorithm for the Voronoi diagram of circles in the Euclidean distance measure based on the idea of edge-flipping has been available recently [17, 18]. In addition, a practical algorithm for Euclidean Voronoi diagram of spheres in 3D, VD(S), by tracing Voronoi edges has been reported very recently [19, 20]. Even though the algorithms for VD(S) have been recently reported, the studies on the data structure for its topology have not been reported yet. As is usually the case with many computational geometry problems, the correctness and computational efficiency of computing VD(S) is heavily dependent on the data structure of its topology. Therefore, an efficient topological representation for VD(S) is needed.

The topology of VD(S) is a cell structure which is a typical non-manifold model. The topology of Euclidean Voronoi diagram for points, VD(P), is also a cell structure. However, the topology of VD(S) is distinguished from the case of points in the sense that a Voronoi face of VD(S) may have a number of holes. Therefore, the topological representations for VD(P) is not directly applicable to VD(S). For the topology of VD(S), we can consider the topological representations for non-manifold geometric models in CAD/CAM. However, VD(S) does not have any non-manifold conditions, such as an isolated vertex, a wire edge and so on, which are found in general non-manifold models in CAD/CAM since the topology of VD(S) is a cell structure. In this paper, the proposed topological representation for VD(S) is a variation of the radial edge data structure with consideration of the topological characteristics of VD(S) distinguished from VD(P) and general non-manifold models in CAD/CAM.

Section 2 of this paper describes an overview of Euclidean Voronoi diagram of spheres including its practical algorithm and applications in biology. Its topological characteristics are discussed in Section 3. Then, the proposed topological representation for VD(S) is presented in Section 4. In Section 5, various queries for the spatial reasoning in the proposed representation are introduced and analyzed. Section 6 presents the analysis of time and storage complexities of the proposed representation. Finally, we conclude this paper with some future works.

*Corresponding author:
Tel: +82 (0)2-2220-0472
Fax: +82 (0)2-2292-0472
Homepage: http://voronoi.hanyang.ac.kr
E-mail: dskim@hanyang.ac.kr

2. Overview of Euclidean Voronoi Diagram of Spheres in 3D

2.1. Definition

Let $S = \{s_1, s_2, \cdots, s_n\}$ be a set of generators for a

Voronoi diagram where $s_i = (c_i, r_i)$ is a 3-dimensional sphere with a center $c_i = (x_i, y_i, z_i)$ and a radius r_i. We assume that no sphere is completely contained inside another sphere even though intersections between spheres are allowed. Associated with each sphere s_i, there is a corresponding *Voronoi region* VR_i where $VR_i = \{p | dist(p,s_i) \leq dist(p,s_j), i \neq j\}$. Then, the set of Voronoi regions $VD(S) = \{VR_1, VR_2, \cdots, VR_n\}$ is called a *Euclidean Voronoi diagram for S* which is defined by the ordinary L_2-metric.

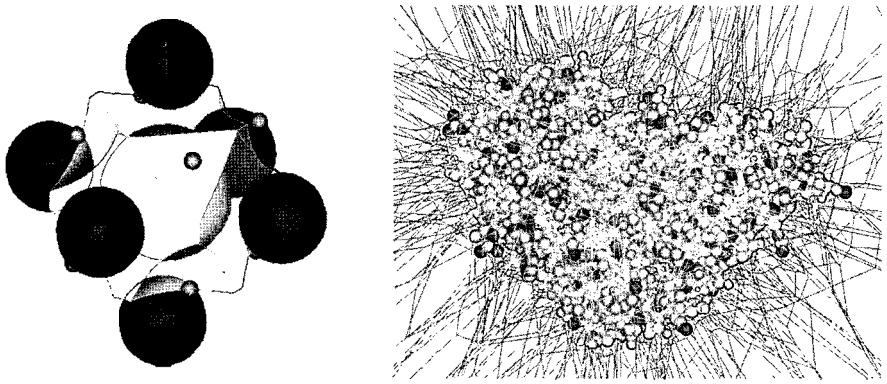
Like an ordinary point set Voronoi diagram, some Voronoi regions corresponding to spheres on the boundary of the convex hull of S are unbounded. Other regions are bounded by a set of faces, called *Voronoi faces*, where a Voronoi face is defined by two immediately neighboring generators. A Voronoi face intersects another Voronoi face to form a *Voronoi edge*, which is a conic section. When Voronoi edges intersect, a *Voronoi vertex* is defined. In this paper, we assume that the degree of a Voronoi vertex is always four which means that no five generators are cotangent to an empty sphere. We also assume that a Voronoi edge is always a common intersection among three Voronoi faces and therefore only three generators define an edge.

Unlike other diagrams, there are only few previous works on VD(S). Aurenhammer discussed the transformation of the computation of VD(S) in d-dimension to that of a $(d+1)$-dimensional power diagram obtained from the convex hull in $(d+2)$-dimensions [2]. Will wrote a comprehensive Ph.D. thesis dedicated to the computation of Voronoi regions in VD(S) [34]. In this work, Will showed that the Voronoi region of a sphere has a $\Theta(n^2)$ combinatorial complexity and proposed the lower envelope algorithm which takes an $O(n^2 \log n)$ expected time for a single Voronoi region, where n is the number of spheres. Luchnikov et al. proposed a practical approach for tracing edges which is a simple yet powerful way to obtain the desired diagram [24]. Gavrilova, in her Ph.D. thesis, reported several properties of VD(S) in arbitrary dimensions, including shapes of the Voronoi regions, nearest neighbors and empty-sphere properties [13, 14]. Boissonnat and Karavelas reported an elegant algorithm to compute a Voronoi region using the convex hull of spheres transformed by inversion [4]. They showed that a Voronoi region can be constructed in $O(n^2)$ time in the worst-case. Recently, Kim et al. reported on the full implementation of the edge-tracing algorithm for constructing a complete Voronoi diagram, not a single region, with discussions on various applications including the analysis of protein structures [19, 20]. They showed that the whole Voronoi diagram can be constructed in $O(n^3)$ time in the worst-case.

2.2. An edge-tracing algorithm

The basic idea of the edge-tracing algorithm is quite simple. The algorithm first locates an initial Voronoi vertex v_0 defined by four appropriate nearby spheres. Provided that v_0 has been found, four edges e_0, e_1, e_2, and e_3 emanating from v_0 can be easily identified and pushed into a stack called an *Edge-stack*. Hence, those edges have v_0 as their starting vertices.

After popping an edge from the stack, the algorithm computes the end vertex of the popped edge. To compute the end vertex, we compute tangent spheres from three spheres defining the popped edge and one of *n-3* candidate spheres. Then, the end vertex is defined by the center of the closest empty tangent sphere to the start vertex of the popped edge among the tangent spheres. Therefore, the end vertex is obtained by *n-3* computations of tangent spheres in the worst case. Once the end vertex of the popped edge is found, it is also possible to define three more edges emanating from this new vertex. Hence, these edges are created and the new vertex is used as the start vertex for three new-born edges. Note that these edges are also pushed into the Edge-stack. The computation of Voronoi



(a) (b)

Fig. 1. Examples of VD(S). (a) the Voronoi region of the centered sphere in fifteen spheres with three different radii, (b) the Voronoi diagram of PDB ID 1EEQ consisting of 1,737 atoms.

diagram of a connected edge graph is completed by following this process until the Edge-stack is empty.

This process iterates as many times as the number of edges m, which can be $O(n^2)$ in the worst-case yet $O(n)$ on the average, where n is the number of spheres. In addition, $O(n)$ time is taken to compute the end vertex for each iteration. Hence, the edge-tracing algorithm constructs VD(S) in $O(mn)$ time in the worst-case. The details of the edge-tracing algorithm can be found in [19, 20].

Shown in Fig. 1 are examples of VD(S). For convenience of visualization, only the Voronoi region for the sphere in the center is shown in Fig. 1(a) while ignoring edges and faces emanating to the infinity. Note that the Voronoi faces in Fig. 1(a) are parts of hyperboloid, and the Voronoi edges are conics represented by rational quadratic Bézier curves. Fig. 1(b) illustrates the Voronoi diagram for a complete protein data with the PDB ID 1EEQ with 1,737 atoms (1,096 C's, 285 N's, 352 O's, and 4 S's) [29]. For convenience of visualization, only Voronoi edges and Voronoi vertices are shown in Fig. 1(b).

2.3. Applications in biology

A protein consists of the permutations of 20 different kinds of amino acids which consist of atoms. In biology, it is known that the geometric structure of protein leads to its functions. Therefore, while early scientific efforts have been mainly focused on the physicochemical aspect of protein, recently more attention has been given to its geometric structure. To analyze the geometric structure of protein, previous studies have mainly used either VD(P) for the centers of atoms [28, 31], the power diagram of atoms [3], or an α-hull [12]. Considering the atoms constituting a protein have different sizes and the Euclidean nature in protein, the tools used in the previous studies are only the approximations to VD(S). Therefore, VD(S) can be an accurate and powerful tool for the analysis of the geometric structure of protein.

A molecular surface, also known as Connolly surface [7, 8], is defined by rolling the spherical probe on protein modeled as a set of spherical atoms [7, 9, 31]. It is well-known that atoms located at the boundary of a protein determine the protein function [7, 8]. Hence, a molecular surface is important in the studies on the protein functions since the surface has a direct relation with exterior atoms. Once a VD(S) is constructed, the computation of molecular surfaces using different probe sizes can be done with the single VD(S). Shown in Fig. 2(a) are the protein (PDB ID: 2LVE) and its molecular surfaces defined by different probes with radii of 2 Å and 8 Å.

Docking between a protein and a small molecule usually occurs around depressed regions, called binding sites or pockets, on the surface of a protein. It is known that the recognition of pockets on proteins is one of the most fundamental processes in drug design. Therefore,

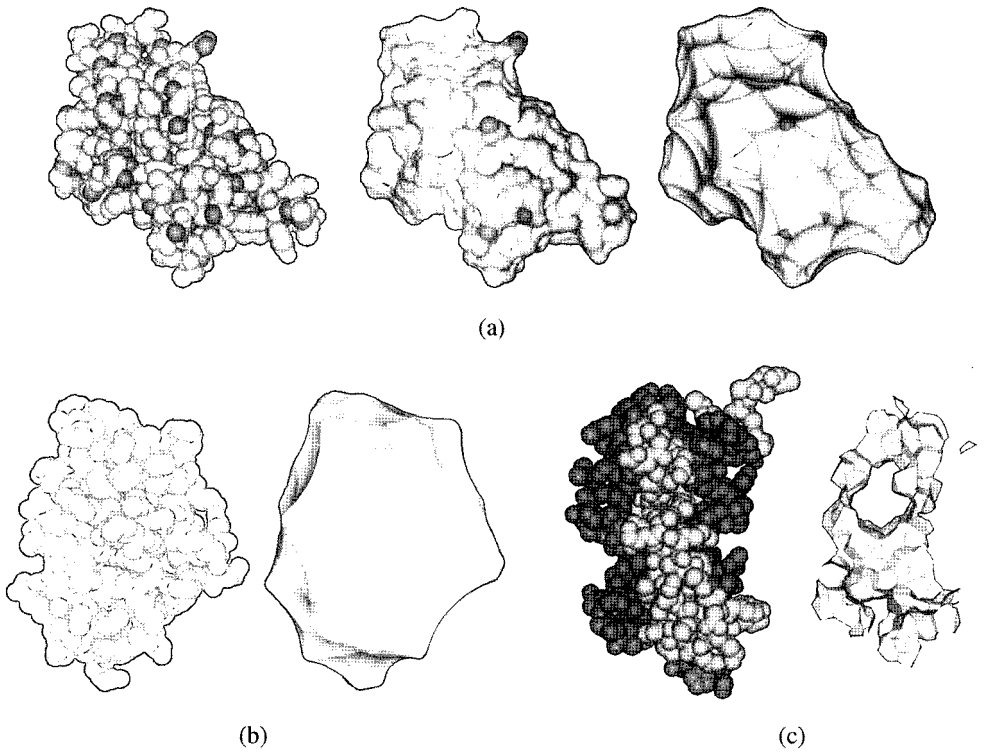


Fig. 2. Applications of VD(S) in protein structure analysis. (a) a protein (PDB ID: 2LVE) and its molecular surfaces by different probes with radii 2 Å and 8 Å, (b) a protein (PDB ID: 1FKF) and its recognized pocket, and (c) a protein (PDB ID: 1AOJ) and its tinteraction interface.

the automatic recognition of pockets and the evaluation of the bindings of a chemical to the pockets are very important in the study on the docking for the development of new drugs. To do this, VD(S) can be a powerful tool. A protein (PDB ID: 1FKF) docked with a small molecular and its automatically recognized pocket is shown in Fig. 2(b).

Amino acids in a protein form chains by being linearly connected to one another via peptide bonds. Since the interaction among chains is critical for the protein functions, the understanding for the interaction is becoming more important. Together with physico-chemical approaches, studies based on the geometric approaches of the interaction are also gaining more attention. We can define the interaction interface as a mid-surface between the two chains in a protein. Shown in Fig. 2(c) are a protein (PDB ID: 1AOJ) which consists of two chains and its interaction interface defined by VD(S).

3. Topological Characteristics of VD(S)

Each Voronoi region in VD(S) is star-shaped with respect to the center of the corresponding spherical generator [27]. Therefore, a Voronoi region itself is a manifold. Each Voronoi region, unless it is an unbounded region, is bounded by a set of Voronoi faces. Two neighboring Voronoi regions share only one Voronoi face on their boundary. Hence, VD(S) is a set of mutually exclusive Voronoi regions and therefore forms a cell structure which is a typical non-manifold model. This means that a non-manifold topological representation should be used for the topology of VD(S).

The topology of Euclidean Voronoi diagram for points in 3D, VD(P), defines a cell structure and therefore VD(P) is also a non-manifold model. Since the dual of VD(P) is a Delaunay triangulation, the topology of VD(P) can be represented as its dual representation which may be very compact and memory efficient. In addition, there have been rigorous studies on the topological representations for VD(P) [5, 11, 23].

However, there are topological differences between VD(S) and VD(P). Even though the dual of VD(S) may form a triangulation for a well-behaving sphere set, it may not be a triangulation in general. As shown in Fig. 3, a small ball is placed between two big balls. Four other balls with identical radii are in the vicinity. Let us consider the Voronoi face defined by the two big balls. The outer boundary of the face is bounded by the Voronoi edges defined by two big balls plus one of the four balls. Note that the Voronoi region of the small ball is bounded by only two Voronoi faces each of which is defined by the small ball itself and one of the two big balls. Then, the Voronoi region of the small ball makes a hole in the face defined by the two big balls as shown in Fig. 3. The Voronoi edge defined by

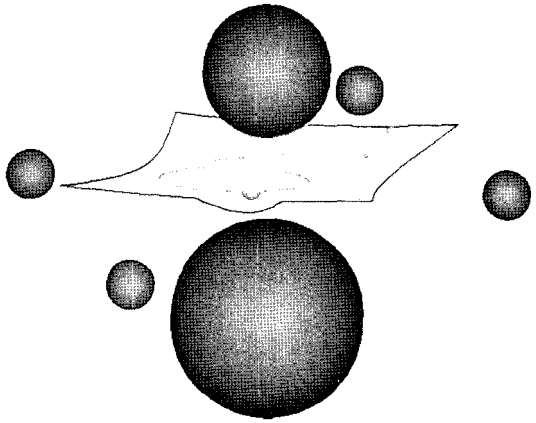


Fig. 3. A Voronoi face with a hole in VD(S) distinguished from the topology of VD(P).

the two big balls and the small ball is the boundary of the hole in the face between the two big balls. Therefore, there is no edge connecting the region of the small ball to any other region. In other words, VD(S) may consist of a number of connected components of edge graphs. Moreover, its dual is not a triangulation, since the edge associated with the hole corresponds to an isolated triangle in its dual. Therefore, the topological representations for VD(P) is not applicable to VD(S).

The data structures for non-manifold geometric models such as radial edge data structure, vertex-based representation, non-manifold topology based on coupling entities, partial entity structure and so on have been extensively studied in CAD/CAM [6, 21, 22, 33, 35]. In the sense that the topology of VD(S) is a non-manifold model, we can consider the employment of these data structures for the topology of VD(S). However, VD(S) is distinguished from general non-manifold models in CAD/CAM.

Since a Voronoi region is star-shaped, a Voronoi region has no disconnected region in itself, it can be represented by only a single shell which is a set of connected boundary faces of a region. In addition, VD(S) does not have any non-manifold conditions such as isolated vertices, wire edges and so on which can be observed in general non-manifold models since VD(S) is a cell structure. Note that an isolated vertex is a vertex which has no incident edges or faces and a wire edge is an edge which has no incident faces. On the other hand, the degrees of vertex and edge are fixed as three and four, respectively. Note that the degree of a vertex is the number of edges incident to the Voronoi vertex and the degree of an edge is the number of faces incident to the Voronoi edge. Hence, VD(S) can be represented by a more compact and simple topological representation than that for general non-manifold models in CAD/CAM. In this paper, we present the topological representation for VD(S) based on the radial edge data structure.

4. Topological Representation Based on Radial Edge Data Structure

The radial edge data structure is an edge based boundary representation for non-manifold models [33]. It is known that there are three kinds of cyclic ordering in non-manifold topology: loop, radial, and disk cycles. Note that the loop cycle is a cycle of edges on the boundary of a face, the radial cycle is a cycle of faces incident to an edge, and the disk cycle is a cycle of edges incident to a vertex. The radial edge data structure extends the manifold representation to the non-manifold representation through a radial cycle. For the storage-efficient implementation of the radial cycle, we employ the partial entity structure of Lee and Lee [22].

The proposed topological representation for VD(S) consists of four *primitive topological entities* and two *supplementary topological entities* as shown in Fig. 4. The primitive topological entities are the *region*, *face*, *edge*, and *vertex*. The supplementary topological entities are *loop* and *partial edge*. Each primitive topological entity has the one-to-one correspondence to its corresponding Voronoi entity and stores corresponding geometric information. Supplementary topological entities maintain the adjacency relationships among primitive topological entities. Fig. 4 shows the overview of the proposed topological representation.

In the primitive topological entities, a region corresponds to a Voronoi region and has a link to its spherical generator. A face for a Voronoi face has a link to its geometry, a hyperboloid. An edge corresponds to a Voronoi edge and also has a link to its geometry which is a conic section. A vertex entity has a spatial position of a Voronoi vertex. The geometric information is not explicitly described in this paper since it is not necessary for the understanding of the topology of VD(S) and the topology and geometry of VD(S) can be separately implemented and maintained.

As shown in Fig. 4, the proposed topology representation consists of 12 adjacency relationships among topological entities and the relationships are represented by arrows. The numbers on the arrows are the number of pointers used for the adjacency relationship between two topological entities. $|F_R|$ and $|L_F|$ denote the number of faces bounding the corresponding region and the number of loops in the corresponding face, respectively.

For the adjacency relationships between a Voronoi region and a Voronoi face, a region points to all the bounding faces, and a face points to two topologically neighboring regions of which spheres define the geometry of the corresponding Voronoi face.

In the case of a Voronoi face and a Voronoi edge, a face is connected with edges on its boundary via loops and partial edges as shown in Fig. 4. A loop is the representation for the outer or inner boundary of a Voronoi face which may have a number of holes. Therefore, a face may point to more than one loop and each loop points to a face. A boundary of a Voronoi face represented by loops can be accessed by following partial edges. To do this, each loop points to one of partial edges on the boundary.

A partial edge plays an important role in loop and radial cycles of VD(S) as a non-manifold model. For the loop cycle, each partial edge points to a loop and its previous and next partial edges on the connecting order as shown in Fig. 5(a). Since a Voronoi edge is a common intersection of three Voronoi faces, an edge is associated with only three partial edges and each partial edge points to the corresponding face via a loop. These three partial edges point to one another for the radial cycle as shown in Fig. 5(b). An edge points to one of the three partial edges and each partial edge points to the edge.

The adjacency relationships between a Voronoi edge and a Voronoi vertex are represented by an edge pointing to its two vertices, and a vertex pointing to all incident edges. The disk cycle is implicitly represented by the loop and radial cycles and the fixed degree of a vertex.

The adjacency relationships can be preserved under the constant orientation of topological entities. In this paper, we assume that the orientations of a Voronoi face and a Voronoi edge are as follows. The orientation of a Voronoi face is defined as the direction from one

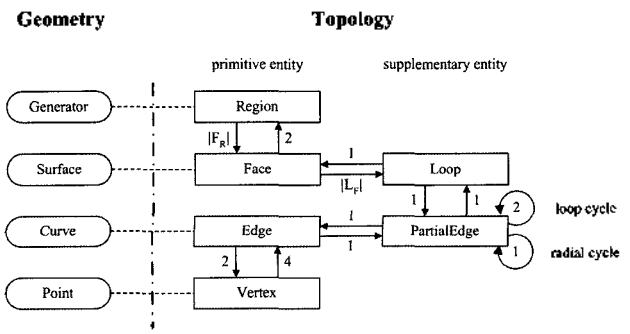


Fig. 4. Overview of topological representation for EVD(S).

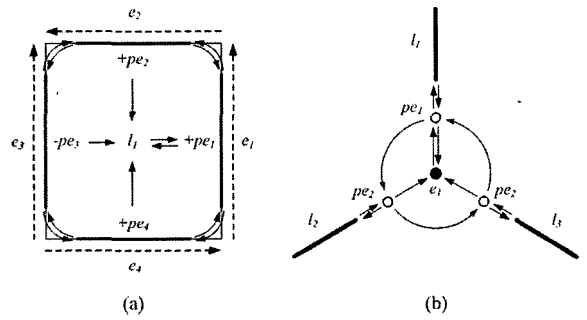


Fig. 5. Adjacency relationships between Voronoi face and edge in loop and radial cycles. (a) the loop cycle by loop and partial edge entities, (b) the radial cycle by partial edge and edge entities.

of two incident regions to the other. The start region and the end region on the orientation of a Voronoi face are denoted by the left region and the right region of the corresponding face, respectively. The orientation of a Voronoi edge is defined as the direction from its start vertex to its end vertex. Then, a set of Voronoi edges bounding a Voronoi face is ordered in CCW orientation as seen from the right region of the face and three Voronoi faces to share a common Voronoi edge are connected in CCW orientation as seen from the end vertex of the common edge.

In our representation, the order of the partial edges in a loop cycle depends on the orientation of the face. In Fig. 5(a), the face with one loop, l_1, bounded by four partial edges, pe_1, pe_2, pe_3, and pe_4, is seen from its right region. Then, each partial edge is connected in CCW orientation. Therefore, pe_1 of l_1, points to pe_2, and pe_4 which are its next and previous partial edges in the loop cycle, respectively. The orientation of an edge determines the order of partial edges in a radial cycle. In Fig. 5(b), the edge e_1 seen from its end vertex has three partial edges, pe_1, pe_2, and pe_3 linked in CCW orientation. For example, the next partial edge of pe_1 in a radical cycle is pe_2.

The relationship between the face orientation and the edge orientation is preserved by partial edges. A partial edge itself is not orientable but has a relative orientation between a Voronoi face and its bounding Voronoi edge. In Fig. 5(a), pe_1 is marked with "+" since e_1 has the identical orientation with the face orientation bounded by the edge. On the other hand, pe_3 is marked with "–" since the orientation of e_3 is opposite from that of the face. That is, the relative orientation of a partial edge is determined by comparing the orientation of its corresponding edge and that of the face. The adjacency relationships and the consistent orientation in the proposed representation enable us to deal with various queries for the spatial reasoning in VD(S).

5. Topological Queries for Spatial Reasoning

As discussed in the previous section, the proposed topological representation for VD(S) consists of 12 adjacency relationships. The region, face, loop, partial edge, edge, and vertex sets are denoted by R, F, L, PE, E, and V, respectively. An adjacency relationship between X and Y is denoted by $X \rightarrow Y$ which means that there is a topologically direct connection from X to Y, where X and Y can be R, F, L, PE, E, and V. For example, $V \rightarrow E$ means that the vertex is directly connected with the edge. Therefore, the topological representation can be thought of as a set of adjacency relationships. Through the adjacency relationships, we can reason the space over VD(S). That is, we can deal with various queries for the spatial reasoning in the Voronoi diagram through the adjacency relationships.

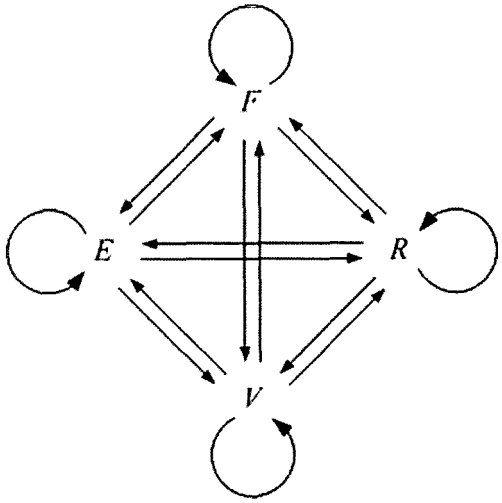


Fig. 6. Primitive topological queries in VD(S).

These queries may be the searches of the neighbor Voronoi entities incident to a given Voronoi entity. There are 16 primitive queries among the four primitive topological entities of region, face, edge, and vertex as shown in Fig. 6.

- Given a vertex, find all neighbor vertices, all incident edges, all neighbor faces, or regions.
- Given an edge, find both extreme vertices, all incident edges or faces, or all neighbor regions.
- Given a face, find all bounding vertices or edges or all neighbor faces or regions.
- Given a region, find all bounding vertices, edges, or faces or all neighbor regions.

$X \Rightarrow Y$ denotes a topological query which finds all neighbor Y entities incident to a given X entity, where X and Y are the sets of each primitive topological entity with the corresponding Voronoi entities. Therefore, X and Y can be R, F, E, and V. For example, $E \Rightarrow F$ means the topological query which finds all incident faces for a given edge. The query can be answered by a *relation sequence* which is the topological linkage of adjacency relationships to connect a given topological entity with a target entity to be found. Shown in Fig. 7 are the relation sequences for 16 primitive topological queries under the schema for the proposed representation.

For example, $E \Rightarrow F$ can be solved by the corresponding relation sequence as shown in Fig. 7. An edge has a direct link to one of its three partial edges. The partial edge points to a loop, and the loop has a link to its corresponding face. Hence, we can find one Voronoi face incident to the given Voronoi edge by visiting the connected partial edge. Then, we can obtain remaining partial edges by traveling along the radial cycle of the given edge so that we can find remaining incident Voronoi faces using these partial edges. Note that in this example, the traversal in a radial cycle can be done by following radial links in

Query	Relation sequence	
$V \Rightarrow V$	$V \rightarrow E \rightarrow V$	$V \Rightarrow E \rightarrow V$
$V \Rightarrow E$	$V \rightarrow E$	
$V \Rightarrow F$	$V \rightarrow E \rightarrow PE \xrightarrow{L} PE \rightarrow L \rightarrow F$	$V \rightarrow E \Rightarrow F$
$V \Rightarrow R$	$V \rightarrow E \rightarrow PE \xrightarrow{L} PE \rightarrow L \rightarrow F \rightarrow R$	$V \rightarrow E \Rightarrow F \rightarrow R$
$E \Rightarrow V$	$E \rightarrow V$	
$E \Rightarrow E$	$E \rightarrow PE \xrightarrow{L} PE \xrightarrow{L} PE \rightarrow E$	
$E \Rightarrow F$	$E \rightarrow PE \xrightarrow{L} PE \rightarrow L \rightarrow F$	
$E \Rightarrow R$	$E \rightarrow PE \xrightarrow{L} PE \rightarrow L \rightarrow F \rightarrow R$	$E \Rightarrow F \rightarrow R$
$F \Rightarrow V$	$F \rightarrow L \rightarrow PE \xrightarrow{L} PE \rightarrow E \rightarrow V$	$F \Rightarrow E \rightarrow V$
$F \Rightarrow E$	$F \rightarrow L \rightarrow PE \xrightarrow{L} PE \rightarrow E$	
$F \Rightarrow F$	$F \rightarrow L \rightarrow PE \xrightarrow{L} PE \xrightarrow{L} PE \rightarrow L \rightarrow F$	
$F \Rightarrow R$	$F \rightarrow R$	
$R \Rightarrow V$	$R \rightarrow F \rightarrow L \rightarrow PE \xrightarrow{L} PE \rightarrow E \rightarrow V$	$R \rightarrow F \Rightarrow E \rightarrow V$
$R \Rightarrow E$	$R \rightarrow F \rightarrow L \rightarrow PE \xrightarrow{L} PE \rightarrow E$	$R \rightarrow F \Rightarrow E$
$R \Rightarrow F$	$R \rightarrow F$	
$R \Rightarrow R$	$R \rightarrow F \rightarrow R$	$R \Rightarrow F \rightarrow R$

Fig. 7. Relation sequences for 16 topological queries.

partial edges denoted by, $PE \xrightarrow{L} PE$. Similarly, $PE \xrightarrow{L} PE$ enable us to travel in a loop cycle. Tracing the relation sequence from a given entity to the target entities is similar to the procedure for the implementation of the query. Therefore, the relation sequences can be a good starting point for the implementation of queries.

Consider another example $E \Rightarrow R$ which finds all neighbor Voronoi regions of a given Voronoi edge. We can obtain two regions by tracing the edge, one partial edge, the loop and the face to the target region. We can also find the remaining neighbor regions by traveling in the radial cycle of the edge. Note that the relation sequence of $E \Rightarrow R$ is a super set of $E \Rightarrow F$ as shown in Fig. 7. This means that we can find all the neighbor regions of a given edge from the result of $E \Rightarrow F$. Therefore, $E \Rightarrow R$ can be represented as $E \Rightarrow F \rightarrow R$. The relation sequences summarized in Fig. 7 show the relationships among primitive queries and can be used for analysis of the queries. From an implementation point of view, there may be two methods: One is to implement all queries individually, and the other is to implement one query by calling another queries already implemented. Even though the former is more efficient, the latter has a benefit for code maintenance.

As discussed earlier, topological queries can be used for spatial reasoning in significant applications from various fields. For example, let us consider the interaction interface of a protein which consists of two groups of atoms as shown in Fig. 2(c). Let $A = \{a_1, \cdots, a_m\}$ and $B = \{b_1, \cdots, b_n\}$ be two chains in a protein, where a_i and b_j are atoms with appropriate centers and radii. The interaction interface between chains A and B is defined as $IIF(A,B) = \{p \mid dist(p,A) = dist(p,B)\}$ where $dist(p, A)$ denotes the minimum Euclidean distance from p to the surfaces of all van der Waals atoms in the set A. Then, IIF(A, B) is a subset of Voronoi faces in EVD(A ∪ B).

Since IIF(A, B) is a subset of Voronoi faces in VD(S) for whole atoms, IIF(A, B) can be easily located by simply checking each Voronoi face if it is defined by the atoms in different groups. Therefore, the approach takes only a linear time with respect to the number of Voronoi faces in the Voronoi diagram. On the other hand, if an initial Voronoi face on the interface is found, we can find Voronoi faces incident to the initial Voronoi face by $F \Rightarrow F$ query and collect only the Voronoi faces on the interface among the incident faces. Then, the remaining Voronoi faces on the interface can be found by repeating $F \Rightarrow F$ queries with respect to the newly found faces. Therefore, if an initial Voronoi face on the interface is found, this approach takes a linear time with respect to the number of the Voronoi faces on the interaction interface.

6. Analysis of Time and Storage Complexities

The proposed topological representation for VD(S) can deal fully and efficiently with the topological characteristics of VD(S). In order to verify the performance of the representation, the time and storage complexities of the proposed representation are derived.

The time complexity of the proposed representation is estimated by measuring the response time of the primitive queries. The time is evaluated as the number of pointer evaluations and statement executions. We assume that pointer evaluation time is equal to statement execution time. The total running time of each query is shown in Table 1. Let X_Y denote the number of X entities incident to a given Y entity. For example, F_R is the number of face entities bounding the given region entity. Note that E_V, R_V, and PE_E are in fact 4, 4, and 3, respectively since the degree of a vertex is four and an edge is always a common intersection among three Voronoi faces. In addition, V_E is 2 since a Voronoi edge is bounded by two vertices. Therefore, nine queries can be done in constant time as shown in Table 1.

The storage cost of the proposed topological representation is estimated with respect to the size of

Table 1. Time complexity of 16 topological queries

	R	F	E	V
R	$4F_R$	F_R	$F_R+L_R+3.5Pe_R+2E_R$	$F_R+L_R+6.25Pe_R+2V_R$
F	$R_F(2)$	L_F+8Pe_F	L_F+2Pe_F	L_F+4Pe_F
E	$5Pe_E(15)$	$3Pe_E(9)$	$5Pe_E(15)$	$V_E(2)$
V	$2+14Pe_E+2R_V(52)$	$11Pe_E(33)$	$E_V(4)$	$3E_V(12)$

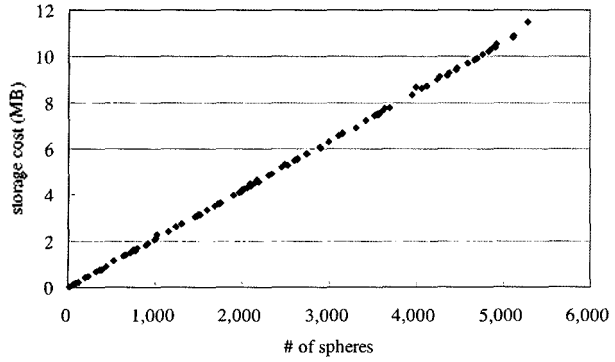


Fig. 8. Storage costs of the proposed representation with respect to the sizes of input data.

the input data. To analyze the storage cost, the following assumptions are made. The storage cost for a pointer is four bytes and the attributes or geometric data for each topological entity are not counted. The storage costs for one hundred protein data from PDB are estimated as shown in the Fig. 8. The figure shows that the storage complexity is linear with respect to the number of spheres.

7. Conclusion

Voronoi diagrams have been known for several important applications in science and engineering. While the properties and algorithms for the ordinary Voronoi diagram of a point set have been well-known, their counterparts for a set of spheres have not been sufficiently studied. Even though the practical algorithms for VD(S) have been reported recently, studies on the data structure for its topology have not yet been reported.

In this paper, we presented the topological representation for VD(S) based on the radial edge data structure. Since VD(S) is a set of mutually exclusive Voronoi regions, it defines a cell structure which is a typical non-manifold model. This means that a non-manifold topological representation should be used to represent the topology of VD(S). Since, however, VD(S) has no such non-manifold conditions as isolated vertices, wire edges and so on which can be observed in general non-manifold models, the proposed topological representation for VD(S) could be more compact and simple than that for a general non-manifold model.

The topological entities for VD(S) and the adjacency relationships among them under the consistent orientation were also discussed. The sixteen primitive topological queries for spatial reasoning over VD(S) could be answered by relation sequences which can be a good starting point for the implementation of queries. Through the analysis of time and storage complexities for the proposed representation, we showed that nine queries can be done in constant time and the storage costs are linear with respect to the size of input data. In the future, we will study more compact data structure which may be obtained by the dual structure such as the Delaunay triangulation as the dual of Voronoi diagram of points.

Acknowledgements

This research was supported by Creative Research Initiatives from Ministry of Science and Technology, Korea.

References

[1] Angelov, B., Sadoc, J.-F., Jullien, R., Soyer, A., Mornon J.-P., and Chomilier, J. (2002), Nonatomic solvent-driven Voronoi tessellation of proteins: an open tool to analyze protein folds, *Proteins: Structure, Function, and Genetics*, **49**(4), 446-456.

[2] Aurenhammer, F. (1987), Power diagrams: properties, algorithms and applications, *SIAM Journal of Computing*, **16**, 78-96.

[3] Bajaj, C.L., Pascucci, V., Shamir, A., Holt, R.J., and Netravali, A.N. (2003), Dynamic maintenance and visualization of molecular surfaces, *Discrete Applied Mathematics*, **127**, 23-51.

[4] Boissonnat, J.D. and Karavelas, M.I. (2003), On the combinatorial complexity of Euclidean Voronoi cells and convex hulls of d-dimensional spheres, Proceedings of the 14th annual ACM-SIAM Symposium on Discrete Algorithms, 305-312.

[5] Brisson, E. (1989) Representing geometric structures in d dimensions: topology and order, Proceedings of 5th Symposium on Computational Geometry, ACM Press, New York, 218-227.

[6] Choi, Y. (1989), *Vertex-based Boundary Representation of Non-manifold Geometric Models*, Ph.D. Dissertation, Carnegie Mellon University, USA.

[7] Connolly, M.L. (1983), Analytical molecular surface calculation, *Journal of Applied Crystallography*, **16**, 548-558.

[8] Connolly, M.L. (1983), Solvent-accessible surfaces of proteins and nucleic acids, *Science*, **221**, 709-713.

[9] Connolly, M.L. (1996), Molecular surfaces: a review, *Network Sci.*

[10] Dafas, P., Bolser, D., Gomoluch, J., Park, J., and Schroeder, M. (2004), Using convex hulls to compute protein interactions from known structures, *Bioinformatics*, **20**(10), 1486-1490.

[11] Dobkin, D.P. and Laszlo, M.J. (1987), Primitives for the Manipulation of Three-Dimensional Subdivisions, Proceedings of 3rd Symposium on Computational Geometry, ACM Press, New York, 86-99.

[12] Edelsbrunner, H., Facello, M., and Liang, J. (1998), On the definition and the construction of pockets in macromolecules, *Discrete Applied Mathematics*, **88**, 83-102.

[13] Gavrilova, M. (1998), *Proximity and Applications in General Metrics*, Ph. D. Thesis, The University of Calgary, Dept. of Computer Science, Calgary, AB, Canada.

[14] Gavrilova, M. and Rokne, J. (2003), Updating the topology of the dynamic Voronoi diagram for spheres in Euclidean d-dimensional space, *Computer Aided*

Geometric Design, **20**(4), 231-242.

[15] Goede, A., Preissner, R., and Frömmel, C. (1997), Voronoi cell: new method for allocation of space among atoms: elimination of avoidable errors in calculation of atomic volume and density, *Journal of Computational Chemistry*, **18**(9), 1113-1123.

[16] Kim, D.-S., Chung, Y.-C., Kim, J.J., Kim, D., and Yu, K. (2002), Voronoi diagram as an analysis tool for spatial properties for ceramics, *Journal of Ceramic Processing, Research*, **3**(3), 150-152.

[17] Kim, D.-S., Kim, D., and Sugihara, K. (2001), Voronoi diagram of a circle set from Voronoi diagram of a point set: I. Topology, *Computer Aided Geometric Design*, **18**(6), 541-562.

[18] Kim, D.-S., Kim, D., and Sugihara, K. (2001), Voronoi diagram of a circle set from Voronoi diagram of a point set: II. Geometry, *Computer Aided Geometric Design* **18**(6), 563-585.

[19] Kim, D.-S., Cho, Y., Kim, D., and Cho, C.-H. (2004), Protein structure analysis using Euclidean Voronoi diagram of atoms, Proceedings of International Workshop on Biometric Technologies (BT 2004), Special Forum on Modeling and Simulation in Biometric Technology, 125-129.

[20] Kim, D.-S., Cho, Y., and Kim, D. (2005), Euclidean Voronoi diagram of 3D balls and its computation via tracing edges, *Computer-Aided Design* **37**(13), 1412-1424.

[21] Lee, K. (1999), Principles of CAD/CAM/CAE Systems, Addison Wesley.

[22] Lee, S.H. and Lee, K. (2001), Partial entity structure: a compact boundary representation for non-manifold geometric modeling, *ASME Journal of Computing & Information Science in Engineering*, **1**(4), 356-365.

[23] Lienhardt, P. (1989), Subdivisions of n-dimensional spaces and n-dimensional generalized map, Proceedings of 5th Symposium on Computational Geometry, ACM Press, New York, 228-236.

[24] Luchnikov, V.A., Medvedev, N.N., Oger, L., and Troadec, J.-P. (1999), Voronoi-Delaunay analysis of voids in systems of nonspherical particles. *Physical Review E*, **59**(6), 7205-7212.

[25] Montoro, J.C.G. and Abascal, J.L.F. (1993), The Voronoi polyhedra as tools for structure determination in simple disordered systems, *The Journal of Physical Chemistry*, **97**(16), 4211-4215.

[26] Ohno, K., Esfarjani, K., and Kawazoe, Y. (1999), *Computational Materials Science From Ab Initio To Monte Carlo Methods*, Springer.

[27] Okabe, A., Boots, B., and Sugihara, K. (1992), *Spatial Tessellations: Concepts and Applications of Voronoi, Diagram*, John Wiley & Sons.

[28] Peters, K.P., Fauck, J., and Frömmel, C. (1996), The automatic search for ligand binding sites in protein of know three-dimensional structure using only geometric criteria, *Journal of Molecular Biology*, **256**, 201-213.

[29] RCSB Protein Data Bank (April 8, 2004), http://www.rcsb.org/pdb/.

[30] Richards, F.M. (1974), The interpretation of protein structures: total volume, group volume distributions and packing density, *Journal of Molecular Biology*, **82**, 1-14.

[31] Richards, F.M. (1977), Areas, volumes, packing and protein structure, *Annu. Rev. Biophys. Bioeng*, **6**, 151-176.

[32] Voloshin, V.P., Beaufils, S., and Medvedev, N.N. (2002), Void space analysis of the structure of liquids, *Journal of Molecular Liquids*, **96-97**, 101-112.

[33] Weiler, K. (1988), The radial edge structure: a topological representation for non-manifold geometric boundary modeling, in: Wozny, M.J., McLaughlin, H.W. and Encarnacao, J.L. Eds., *Geometric Modeling for CAD Applications*, North Holland, Elsevier Science Publishers B.V., 3-36.

[34] Will, H.-M. (1999), *Computation of Additively Weighted Voronoi Cells for Applications in Molecular Biology*, Ph.D. Dissertation, Swiss Federal Institute of Technology, Zurich.

[35] Yamaguchi, Y. and Kimura, F. (1995), Nonmanifold topology based on coupling entities, *IEEE Computer Graphics and Applications*, **15**(1), 42-50.

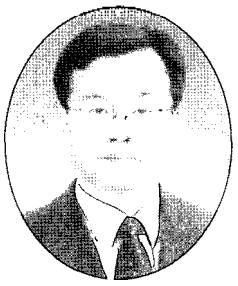
Youngsong Cho is a research assistant professor in Voronoi Diagram Research Center at Hanyang University, Korea. He received his B.S., M.S., and Ph.D. degrees from Hanyang University, Korea in 1995, 1997, and 2003, respectively. His current research interests are the theories and applications of Voronoi diagram and geometric modeling.

Donguk Kim is a research assistant professor in Voronoi Diagram Research Center at Hanyang University, Korea. He received his B.S., M.S. and Ph.D. degrees from Hanyang University in 1999, 2001 and 2004, respectively. His research interests include computational geometry, geometric modeling and their applications in the molecular biology.

Deok-Soo Kim is Professor in Department of Industrial Engineering at Hanyang University, Korea. He is also currently Director of Voronoi Diagram Research Center supported by Ministry of Science and Technology, Korea. Before he joined the university in 1995, he worked at Applicon, USA, and Samsung Advanced Institute of Technology, Korea. He received a BS from Hanyang University, Korea, an MS from the New Jersey Institute of Technology, USA, and a Ph.D. from The University of Michigan, USA, in 1982, 1985 and 1990, respectively. His current research interests are the theories and applications of Voronoi diagram and geometric modeling.

Youngsong Cho

Donguk Kim



Deok-Soo Kim