Computation of Molecular Surface Using Euclidean Voronoi Diagram

Joonghyun Ryu[2], Donguk Kim[3], Youngsong Cho[4], Rhohun Park[5] and Deok-Soo Kim[1]

[1]Hanyang University, dskim@hanyang.ac.kr
[2]Voronoi Diagram Research Center, jhryu@voronoi.hanyang.ac.kr
[3]Voronoi Diagram Research Center, donguk@voronoi.hanyang.ac.kr
[4]Voronoi Diagram Research Center, yscho@voronoi.hanyang.ac.kr
[5]Voronoi Diagram Research Center, rhpark@voronoi.hanyang.ac.kr

ABSTRACT

Given a protein, analyzing the geometric structure of protein is fundamental for the study of a protein folding, docking, interactions between proteins, and so on. One of the important geometric analyses is computing the molecular surface of protein. Discussed in this paper is an efficient algorithm to compute such a molecular surface of protein via the concept of blending operation among atoms constituting the protein. To facilitate the decision for the existence of blending surface among atoms, we take advantage of the proximity information of Euclidean Voronoi diagram of atoms. The proposed algorithm initially detects topological locations where blending surface exist via both edge accessibility and face accessibility of Euclidean Voronoi diagram.

Keywords: Protein structure, Molecular surface, Blending surface, Euclidean Voronoi diagram

1. INTRODUCTION

A protein consists of amino acids, and an amino acid consists of atoms. Hence, a protein usually consists of from thousands to hundreds of thousands of atoms. Fig.1 shows van der Waals surfaces of atoms for a synthetic human immunoglobulin VL lamda domain downloaded from Protein Data Bank (PDB) [18]. Each protein entry in PDB is associated with an id and the protein shown in the figure has an id of 1BH8 and consists of 1074 atoms. Given a protein, it is often necessary to study its geometric and physicochemical characteristics for the engineering of proteins.

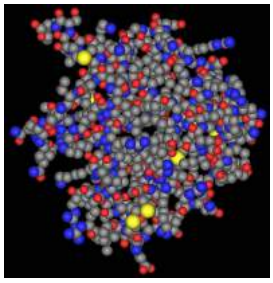


Fig. 1. Union of 3D spheres for molecules

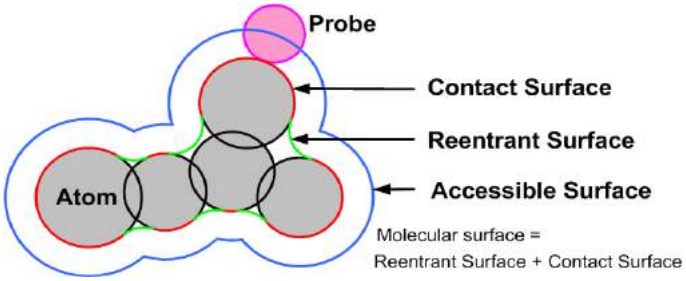


Fig. 2. Solvent accessible vs. Molecular surface

As we better understand the definition and manipulation of geometry using computers, there have been efforts to apply our knowledge on geometry to biosystems such as proteins, DNA, RNA, etc. It turns out that geometric molecular modeling plays important roles in the prediction of protein functions, drug design, simulation, etc.
Discussed in this paper is an algorithm to compute molecular surfaces, which is known as one of useful geometric tools in molecules, assuming a convenient computational tool for the proximity among constituting atoms is available. The proposed algorithm initially identifies the appropriate topological locations where the molecular surface exists and then computes its surface equations. In our research, the Voronoi diagram is used for the efficient search for the neighborhood information. In particular, we have devised an algorithm and fully implemented the Voronoi diagram of atoms in Euclidean metric, Euclidean Voronoi diagram (EVD) for short [10], [11].

2. PREVIOUS WORKS

Due to its importance, there have been several researches on the geometric structure of proteins. Lee and Richards first defined the concept of solvent accessible surface to compute the free space that a probe, a sphere enclosing a small molecule, can move around during the small molecule interacts with a protein [12]. Among others, Connolly defined a molecular surface to calculate the volume of protein, electrostatic potential, interface surfaces in protein-protein, protein-ligand, etc [4], [7]. The initial algorithm by Connolly for computing molecular surface is to calculate the sampled points on molecular surface. The sampled surface was rendered as fields of dots because of the deficiency of computing power [6], [15]. The first analytic surface representation is also provided by Connolly [4]. He categorized the patches of molecular surface into three types: Convex spherical patch, saddle-shaped toroidal patch and concave spherical patch. The triangulation of the molecular surface is also discussed in [5], [7]. Seidl and Kriegel presented the algorithm for computing molecular surface by calculating and building up the multi-arc contours on the van der Waals spheres [21]. Varshney et al. discussed a fast computation of molecular surface so that the structure analysis can be done interactively [22]. The computation enhancement was due mainly to the efficient search of neighbors among atoms using the power diagram of atoms constituting the protein.

Bajaj et al., on the other hand, presented a nice treatise for trimmed exact NURBS representation of molecular surface so that a standard graphics library such as OpenGL can be conveniently used without any additional effort [1]. It seems that NURBS representation may facilitate proteins to be modeled within contemporary commercial CAD/CAM softwares. Bajaj also used a power diagram for the efficient neighborhood search among atoms defining blending surfaces of probe sphere. Recognizing the limitation of power diagram, he discussed the conditions for the topology of power diagram to be fixed and an efficient update of correct topology of power diagram if needed, where the radius of solvent is continuously modified [1], [2]. Recently, Edelsbrunner et al. introduced the concept of molecular skin surface, which is the implicit surface defined by the envelope of an infinite family of spheres controlled by a finite collection of weighted points [3], [8]. They also presented the algorithm for triangulating the molecular skin surface in [3].

Review of literature reveals that there are two issues to be observed regarding on the molecular surface. Firstly, the mathematical and computational representation and manipulation of molecular surface itself. Secondly, the use of efficient data structure for the proximity or neighborhood information among atoms so that the queries about nearby atoms defining blending surfaces can be correctly and efficiently located. It seems that the first issue has been pretty much settled down in both implicit and parametric representation of the surface. Regarding on the second issue, however, researchers have mainly used either an ordinary Voronoi diagram of center points of atoms, power diagram of atoms, or α-hull. Considering the atoms constituting a protein may have varying sizes, these approaches only provide close approximations to what is actually and desperately needed: Euclidean Voronoi diagram of atoms, EVD for short. It should be noted that the computation of Euclidean Voronoi diagram for spheres could have been computed recently [10], [11]. Note that the discussions on the computation of a Voronoi cell of Euclidean Voronoi diagram for spheres is also provided by Will [23].

3. CONSTRUCTION OF MOLECULAR SURFACE

3.1 Geometric Model of Molecular Surface

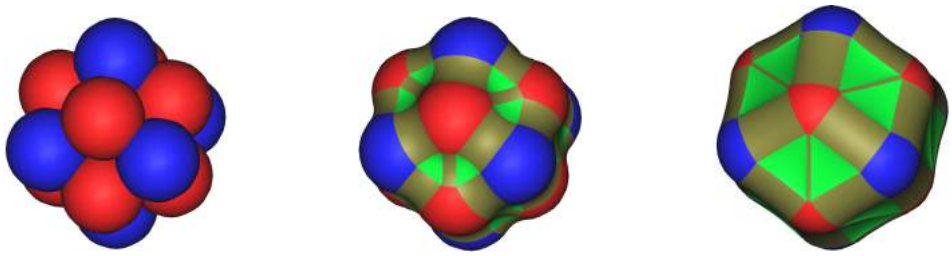
A protein is usually modeled as a set of hard spheres representing atoms, which is called a space-filling or CPK-model, where radii are the van der Waals radii of atoms [6], [12]. Given a CPK-model, there are usually two kinds of surfaces involved as shown in Fig. 2: solvent accessible surfaces and molecular surfaces. A solvent accessible surface, which is first defined by Lee and Richards in 1971[12] and illustrated as a blue curve in the figure, is the set of centers of a spherical probe rolling around the protein. A probe is used for the computational convenience of a small molecule which interacts with the protein. Hence, the solvent accessible surface provides information on the free space that a small molecule can move without penetrating the protein [12]. Note that the notion of solvent accessible surface is similar to the concept of configuration space in robotics community [13] and in fact the solvent accessible surface is simply an offset surface of a given structure in the geometric modeling community [14]. A water molecule is frequently used as a solvent around a protein and the corresponding probe is approximated by a sphere with a radius of 1.4 A [6], [15], [19]. A molecular surface, which is also known as Connolly surface after the name of first researcher defined the surface[4], [7], consists of the most inward points on the probe toward the interior of protein when the probe is in contact with the protein [4], [6], [19]. In other words, the probe can be considered to roll over in any possible direction along the union of atoms without interfering the interior of atoms. The entire envelope created by this probe is, therefore, a molecular surface of the protein. Note that the locus by the centers of the probe contacting the protein is the solvent accessible surface.

A molecular surface again consists of two further categorized surfaces: solvent contact surface and reentrant surface. A solvent contact surface consists of points on the van der Waals atoms which can be contacted by a probe while a reentrant surface is defined as a set of points on the inward part of the surface of a probe sphere, where the probe is located in tangential contact with atoms [19]. The red and green curves in the figure indicate both a solvent contact and a reentrant surface, respectively.

It is quite well-known that atoms located in exterior part of a protein determine the function of the protein [4], [7]. Hence, a molecular surface is important in the study of functions of proteins since the surface has direct relation with exterior atoms. There are two major uses of molecular surface [4], [6], [7], [19]: it completely removes the van der Waals surfaces of interior atoms since these atoms are not directly involved in molecular interactions and most of the van der Waals surfaces of a protein are in the interior. Another use is the visual smoothing of crevices and pits.

It turns out that a reentrant surface, which is a blending surface among atoms of possibly different sizes, again consists of two types of blending surface patches as we call: a rolling blending patch and a link blending patch. We can compute two types of blending surface patches by a ball of the given radius rolling along the union of sphere set in any possible direction with moving tangentially to the union of sphere set. A rolling blending patch is the surface patch generated by a probe rolling along two spheres with moving tangentially to both of them, which is called to be saddle-shaped toroidal patch in the previous work [4], [7]. A link blending patch, denoted as a concave spherical patch [4], [7], is created in the positions where a probe tangentially touching three spheres.

In addition, we define each surface patch constituting a solvent contact surface as an atomic contact patch, which is called convex spherical patch in the previous researches [4], [7]. Thus, a molecular surface consists of the above-mentioned three types of surface patches, as illustrated in Fig. 3. If we roll over the union of same 3D sphere set with probes of different size, we will obtain the molecular surface with different shapes for each case. Note that the radius of the probe for Fig. 3 (c) is approximately ten times larger than that of the probe for Fig. 3 (b).



(a) (b) (c)

Fig. 3. Molecular Surfaces by different probes.(a) a molecule consisting of 15 atoms, (b) defined by smaller probe, (c) defined by relatively larger probe.

3.2 Euclidean Voronoi Diagram of 3D Spheres

The proposed algorithm initially recognizes the topological locations where blending surface patches should appear via edge propagations, which use the accessibility of both edges and faces of EVD. Then, the algorithm computes the geometry counterpart of rolling blending and link blending patches exactly together with their trimming boundary curves. Since the algorithm needs to query neighborhoods of atom spheres to decide topological locations, we briefly describe Euclidean Voronoi diagram of spheres, which can provide us with the answers for such a topological inquiry efficiently.

Despite of many important applications in various disciplines from science and engineering, Euclidean Voronoi diagram for 3D spheres, also known as an additively weighted Voronoi diagram, has not been studied as much as it deserves and its construction algorithm didn't appear in the literatures until recently [10], [11].

The proposed algorithm employs Euclidean Voronoi diagram of 3D sphere set as a supplemental data structure. If Euclidean Voronoi diagram of 3D spheres is given, the various geometric processing such as the computation of molecular surface can be conveniently achieved. Fig. 4 (a) illustrates Euclidean Voronoi diagram of fifteen spheres with three different types of radii and Fig. 4 (b) shows another example for EVD of the same sphere set as in Fig.1.

In this paper, we assume that Euclidean Voronoi diagram of sphere set is already computed and the proximity information is available from EVD of sphere set. In this subsection, we briefly describe the concept and a several basic properties of EVD of 3D spheres which are necessary to develop the proposed algorithm. Let $S = \{s_1, s_2, s_3, \cdots\cdots, s_n\}$ be a set of 3D spheres, which is called generators of Euclidean Voronoi diagram.

$\mathbf{c}_i = (x_i, y_i, z_i)$ and r_i are the center and the radius of each sphere. Then, a Voronoi cell of each sphere is defined as $V(s_i) = \left\{\mathbf{x} \middle\| \|\mathbf{x} - \mathbf{c}_i\| - r_i \le \|\mathbf{x} - \mathbf{c}_j\| - r_j \ for\ j \ne i\right\}$ where Euclidean l_2-norm is used for the distance between 3D points. Then, Euclidean Voronoi diagram of 3D sphere set is the collection of Voronoi cells as follows: $VD(S) = \{V(s_1), V(s_2), V(s_3), \cdots\cdots, V(s_n)\}$

(a) (b)

Fig. 4. Voronoi diagram of 3D spheres.(a) EVD for 15 spheres with 3 different types of radii, (b) EVD for spheres of Fig. 1

We comment two basic assumptions and properties, whose some is necessary for defining each patch constituting blending surfaces and for deciding its topological locations. We assume that every Voronoi vertex in VD(S) has exactly four incident Voronoi edges and the number of the generators defining a Voronoi edge in VD(S) is three. According to the definition of VD(S), every Voronoi vertex in VD(S) is the center of empty 3D sphere tangent to neighboring four generators (spheres). Note that the empty tangent sphere of every Voronoi vertex is computed during the construction of VD(S). For detailed descriptions, refer to references [9], [11] and [16]. In addition, Voronoi edge is conic curve, which can be represented by rational quadratic Bézier curve and every Voronoi face is a hyperboloid. Note the number of the generators defining Voronoi faces of VD(S) is always two.

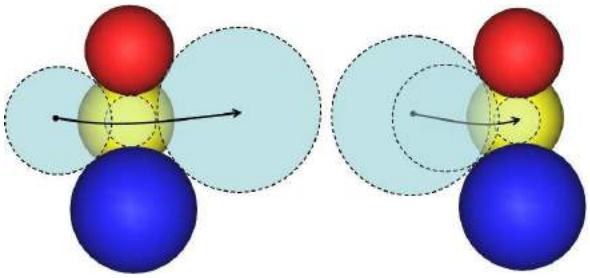
3.3 Existence of Blending Surfaces

Since the blending surface patches should be constructed not inside but only on the molecular surface, it is required to recognize the appropriate locations where blending surface patches are to create. The proposed algorithm locates such positions via edge propagations, which employs the accessibility of an edge and a face of VD(S). Two types of edge and face accessibility are defined as follows:

Definition 1 Edge Accessibility
If a probe with a fixed size can fly freely through all points on an edge with its center placed on the edge, the edge is called fully accessible for the probe. If an edge is partially accessible, the probe can fly through only the proper subset of the edge. The probe cannot be freely positioned on any point of a non accessible edge.

Since there exist its defining three spheres around a Voronoi edge, the edge accessibility can be identified by comparing the radius of a probe with the radius of empty spheres simultaneously tangent to three spheres. Although the number of possible empty tangent spheres is infinite, we can locate empty tangent spheres whose radii are either of the maximum or of the minimum value among the possible empty spheres. Thus, the edge accessibility can be implemented by computing both minimum and maximum empty tangent spheres. Fig.5 (a) and (b) illustrate the possible locations of both maximum and minimum empty tangent spheres. In this figure, illustrated are also two types of Voronoi edges, which are both a monotonic edge and non monotonic edge with regard to the gradation of the radius of empty tangent spheres. Fig.5 (a) shows that a minimum empty tangent sphere occurs on one of intermediate points of the edge while a maximum empty tangent sphere appears on end vertex. Maximum and minimum empty tangent spheres appear on start and end vertex, respectively in Fig. 5(b). Note that such two types of spheres are already computed during the construction of VD(S). Therefore, if the radius of a probe is smaller than minimum value, the corresponding edge is called fully accessible. If the radius of a probe is larger than maximum value, the edge is non accessible. Otherwise, the edge is partially accessible. Note that a minimum empty tangent sphere can be placed on

either one of two vertices or one of intermediate points of the edge while a maximum empty tangent sphere can be placed on either a start or end vertices.



(a) (b)

Fig. 5. Empty tangent spheres of an edge.(a) non monotonic edge case, (b) monotonic edge case w.r.t. the gradation of radii of empty tangent spheres

The accessibility of a face is defined by both the accessibility of its boundary edges and the minimum Euclidean distance between its two defining generators (spheres) as follows:

Definition 2 Face Accessibility
If all of its boundary edges are fully accessible and the minimum distance between two spheres is larger than the diameter of a probe, the face is fully accessible. If either of two conditions is violated, the corresponding face is partially accessible. If all the boundary edges are non accessible, the face is non accessible.

Starting at each unbounded Voronoi edge, we can determine the accessibility of every Voronoi edge via edge propagations, progressively. If the radius of a probe is smaller than minimum radius of current edge, the accessibility of the edge is recognized to be fully accessible and we continue to check the accessibility of incident edges of current edge via current edge propagation. Otherwise, after we determine the accessibility of current edge, where the accessibility is either partially accessible or non accessible, we do not check the accessibility of incident edges any more and terminate current propagation. If this procedure is repeated for each unbounded Voronoi edge, we can determine the accessibility of every Voronoi edge. After then, we can also determine the accessibility of faces based on that of its boundary edges and the minimum distance between its defining two spheres.

Once the accessibility of every Voronoi edge and face is identified, it is possible to compute the precise locations where blending surface patches are defined and the following lemma holds (The proofs are obvious.).

Lemma 1
(Existence of Rolling Blending Patch) If a Voronoi face is partially accessible, there always exists a rolling blending patch between its two defining spheres.
(Existence of Link Blending Patch) If a Voronoi edge is partially accessible, there always exists a link blending patch among its three defining spheres.

3.4 Rolling Blending Surface Patch

A rolling blending patch is the blending surface patch generated by two spheres which define a partially accessible Voronoi face and is constructed by a probe rolling over a gap between two spheres with moving tangentially to both of them (from Lemma 1). A rolling blending patch is the saddle-shaped part of a toroidal surface, whose topology is rectangular. Fig. 6 and Fig. 8 show the constructed rolling blending patches, link blending patches and each atomic contact patch. Atomic contact patches are the remaining portions of spherical surfaces representing atoms from which all of the involved rolling blending patches are removed. Note that there always exist three rolling blending patches on the neighborhood of a link blending patch as shown in Fig.8.

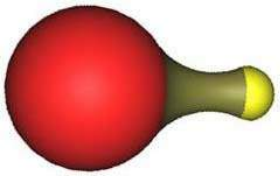


Fig. 6. Rolling blending patch

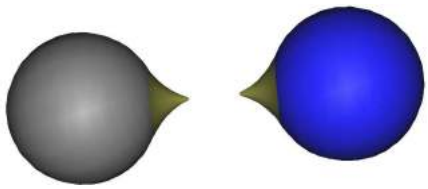


Fig. 7. Self-intersected rolling blending patch

For each pair of spheres which define partially accessible Voronoi faces, the algorithm computes rolling blending patches. In this work, a rolling blending patch is calculated by the surface of revolution. Since the angle necessary for the probe to be rotated with moving tangentially to two spheres also can be computed, we can compute a rolling blending patch by revolving the arc, which is computed from contact points among the probe and two spheres and is denoted by generatrix in the literature [17], around the axis which is defined by connecting two centers of two spheres. The constructed rolling blending patch can be represented as NURBS surface only if we perform the revolution with the generatrix represented as NURBS curve [17]. If the above procedure is performed repeatedly for each pair of spheres defining a partially accessible Voronoi face, the construction of rolling blending patches is completed. Although the intersections between different rolling blending patches does not occur, the self-intersection is possible and in that case, we should compute two revolved surfaces, separately, as illustrated in Fig. 7 [1].

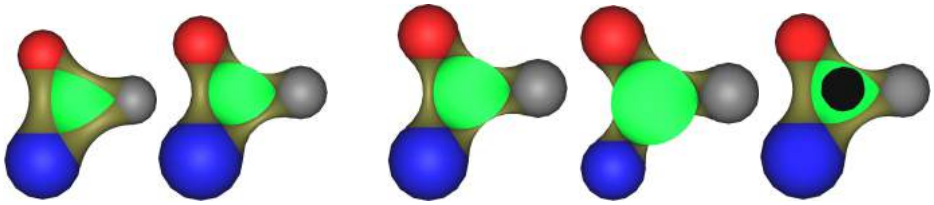
3.5 Link Blending Surface Patch

A link blending patch is the blending surface patch to be present among three spheres which define a partially accessible Voronoi edge and is created by making a probe touch three spheres tangentially and simultaneously (from assumptions of VD(S) and Lemma 1).

If a link blending patch has no intersections with other link blending patches, its topological shape is triangular, as shown in Fig. 8 (a) and its boundary curves consist of three arcs of great circles on a probe sphere. The center of each arc is the center of the probe simultaneously tangent to three spheres (atoms) and two extreme points of each arc are two points among three contact points of a probe with three spheres.

In fact, it is known that there exist no intersections between rolling blending patches and link blending patches or between link blending patches and atomic contact patches [1]. Furthermore, a link blending patch admits no self-intersections [1]. Thus, it is sufficient that we should consider intersections between different link blending patches, which can be classified into two cases. Only if even one of its three neighboring rolling blending patches is self-intersected, the topological shape of a link blending patch is no more triangular. According to the number of self-intersected rolling blending patches, 3, 5, 7 and 9-sided link blending patches can be created as shown in Fig. 8. We can see the occurrence of intersections between the neighboring link blending patches with a self-intersected rolling blending patch. Another case is that a hole is created by the intersection between non-neighboring two link blending patches (Refer to Fig. 8 (e)).

Since a rolling blending patch has only self-intersection without holes, it can be represented as NURBS surfaces without any additional efforts. A link blending patch can have various topological shapes and have even holes via intersections between different link blending patches. Thus, if the parametric representation is introduced, a trimming operation is inevitable and the handling of arbitrary topologies is also needed. In this work, we implicitly represent a link blending patch by maintaining only its boundary curves. Since link blending patches can be transformed to a single-valued function by projecting patches onto a plane defined by three contact points of a probe with three spheres, its tessellations for visualization is not so difficult.



(a) (b) (c) (d) (e)

Fig. 8. (a) 3-sided blending patch, (b) 5-sided blending patch, (c) 7-sided blending patch, (d) 9-sided blending patch (e) hollow patch

Let us consider the algorithm for calculating intersections between different link blending patches, which are spherical triangular patches, as previously described. Fig. 9 shows that the possible intersections between link blending patches. Note that an intersection curve is either a circular arc or a circle because the intersection between 3D sphere/3D sphere is a circle and the intersection between link blending patches is its subset. The proposed intersection algorithm maintains the boundary curves of a trimmed link blending patch, which consists of three great circular arcs and additional trimming arcs, by calculating the trimming arcs. At first, the algorithm locates all the relevant trimming arcs of the target link blending patch. Then, the final boundary curves are constructed by sorting three great circular arcs and trimming arcs.

(a) (b) (c)

Fig. 9. Intersections between link blending patches. (a) No intersection, (b) Intersection with circular arc boundary, (c) Single hole

Let ST_i and ST_j be two spherical triangles, which correspond to two link blending patches and both P_i and P_j be two instances of probes(spheres) which include both ST_i and ST_j, respectively. Assume that P_i is the instance of the probe in which a target link blending patch to be trimmed is embedded. If $P_i \bigcap P_j \neq \phi$, we can find the plane, H such that $P_i \bigcap P_j = P_i \bigcap H$. Then, our intersection problem between ST_i and ST_j is reduced to the intersection between ST_i and H. Since the intersection between sphere and plane is a circle, we should calculate either a circle or a circular arc which is the subset of the intersection between P_i and H. If the intersection between ST_i and ST_j is not a hole inside ST_i, two extreme points of the intersecting circular arc always appear either on one or two among three great circular arcs.

Thus, the problem is again simplified as the intersection between circular arcs and a plane, which can be formulated as a quadratic equation with both an implicit form equation of the plane and a parametric form equation of the arc. The real roots of the equation are the parameter values of intersection points. If the equation has no real root, we should check another possible case, where the intersection occurs inside the spherical triangle as a hole. To that end, we should check whether there exists an intersection between a triangle defined by three contact points of ST_i and a ray defined by both the center of P_i and the center of a hole or not. The existence of the intersection indicates a trimming circle inside ST_i as a hole.

Once all the relevant trimming arcs and circles of ST_i are computed, we should orientate all the arcs consistently, as shown in Fig. 10. If this process is performed repeatedly for the entire possible intersecting link blending patches, the correct final boundary curves can be obtained. Note that both a trimming arc and a trimming circle (a hole inside the patch) appear in Fig. 10. In this case, CCW orientations are assigned for the outer loop boundary curves and CW for the inner loop (a hole). If we consider these situations in a parametric domain, which can be constructed based on a stereographic projection, the previous trimming process can be also explained as the problem to maintain a 2D dynamic union of balls [1].

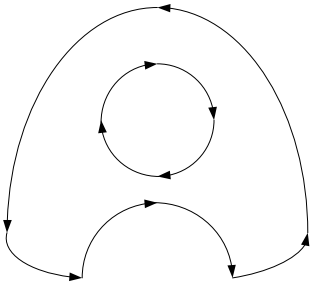


Fig. 10. Orientation of boundary curves of link blend

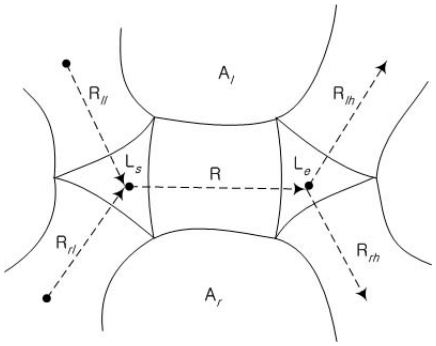


Fig. 11. Rolling blending patch centric topology

3.6 Atomic Contact Surface Patch

An atomic contact patch is the leftovers of a spherical surface from which the regions occupied by its involved rolling blending patches are removed as previously illustrated. Note that only one rolling blending patch is involved for each atom in both Fig. 6 and Fig. 7 and two rolling blending patches are involved for each atom in Fig. 8.

An atomic contact patch is a convex spherical surface and its boundary curves are composed of a set of either circular arcs or circles, which are border between an atomic contact patch and an involved rolling blending patch. Note that the boundary curves of an atomic contact patch in Fig. 6 and Fig. 7 consist of a single circle and those of patches in Fig. 8 consist of two circular arcs. In our approach, an atomic patch is represented implicitly by maintaining its boundary curves, which are locus by the extreme points of generatrix used for computing its involved rolling blending patches (See 3.4).

3.7 Maintenance of Inter-patch Topology

In the application of molecular biology such as protein docking, the local shape of 3D structure of the molecules as well as its physicochemical properties plays important roles for docking retrieval. In this case, the neighborhood query of the given patch is required for either the approximation of the local shape or the segmentation of a molecular surface [20]. The neighborhood query can be facilitated by maintaining the topological information among the surface patches of a molecular surface [20].

A rolling blending patch is always created between two link blending patches and three rolling blending patches are connected to a link blending patch, as shown in Fig. 8. Given a rolling blending patch, there exist two related atomic contact patches, which are constructed on two spheres defining a partially accessible face. Thus, we can define and maintain the topology information among patches based on a rolling blending patch as illustrated in Fig. 11. In Fig.11, R, L and A symbolize a rolling blending patch, a link blending patch and an atomic contact patch, respectively. From the topological relationships among R, L and A, we can construct the planar graph, whose vertices and edges are link blending patches and rolling blending patches, respectively.

In Fig. 11, if we consider R to be an edge which is defined by both Ls and Le as its two vertices, its incident four edges Rlh, Rrh, Rll and Rrl also can be defined. This situation is described by the dotted arrows in that figure. Since two neighboring atomic contact patches of R, Al and Ar can be assumed to be left face and right face of R, following these analogues, we can store the inter-patch topology information into full winged edge data structure. Several alternatives for storing these topological information and the issue of storage requirement are discussed in [20].

4. FAST DETECTION OF INTERSECTIONS BETWEEN BLENDING PATCHES

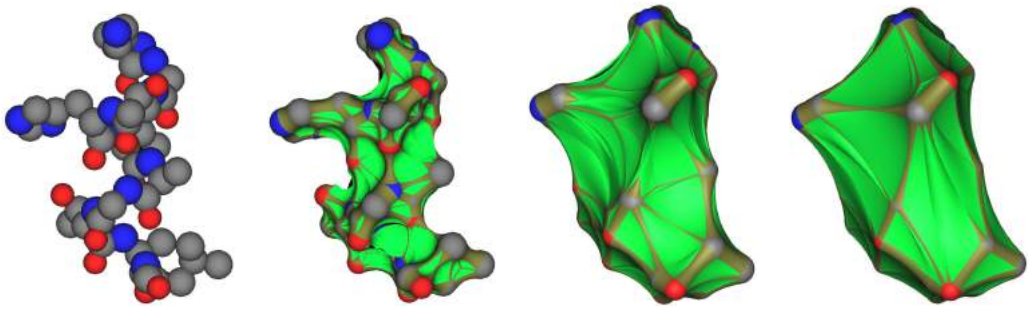
The naive approach for finding the intersections between link blending patches is to solve directly the intersection problems between all of the possible pairs of link blending patches. For reducing the redundant computational burdens, we can filter out non-intersecting pairs before we compute intersections. To that end the algorithm exploits a Voronoi diagram of the probes postured on their appropriate link blending positions [1]. Note that such positions were already identified via edge propagations.

Suppose that we put the spheres of the same radius as that of a given probe on appropriate link blending positions obtained from edge propagations. Then, the Voronoi diagram of probes (spheres) can reduced to a point set Voronoi diagram, because the radius of the probe is fixed.

Let $L = \{L_1, L_2, L_3, \cdots\cdots, L_n\}$ and $P_L = \{P_{L_1}, P_{L_2}, P_{L_3}, \cdots\cdots, P_{L_n}\}$ be a set of link blending patches and a set of corresponding probe instances positioned at each link blending patch. Assume that L_i be current target link blending patch trimmed by its neighboring link blending patches. Suppose P_{L_i} and P_q to be a sphere including L_i and one of its neighboring spheres, respectively. For filtering out the non-intersection cases, the algorithm queries the neighboring spheres of P_{L_i} based on a point set Voronoi diagram, VD(P_L). Then, for our intersection problem, we could exclude the spheres of the radius whose addition with the radius of P_{L_i} is smaller than the distance between two centers of P_{L_i} and P_q. In this way, we can screen out all of non-intersecting neighboring generators and reduce the computation cost.

5. DISCUSSIONS AND CONCLUSION

This paper discusses the construction of molecular surface with being supplemented with Euclidean Voronoi diagram of 3D sphere set. The algorithm initially finds the topological locations of blending surfaces via edge propagations. Then, the algorithm computes the geometry counterpart of rolling blending and link blending patches exactly. The geometry of rolling blending surface patch is represented by NURBS surface via the surface of revolution and that of link blending surface patch is represented implicitly by maintaining only boundary curves of the trimmed link blending patch. Finally, we discussed the maintenance of the inter-patch topology information for three types of constituent patches of a molecular surface. While Bajaj et al. point out the conditions where the additional computation for power diagram is necessary [2], the previous researches based on power diagram requires power diagram of molecules to be updated for reflecting changes in the radius of a probe. However, since the Euclidean Voronoi diagram of 3D spheres is fixed irrelevant to the size of a radius, our algorithm can efficiently compute a molecular surface even when the radius of a probe is continuously modified without updating EVD as illustrated in Fig. 12.



(a) (b) (c) (d)

Fig. 12.(a) original molecule and molecular surfaces for the probes with a diverse radius size, (b) 1.7 Å (c) 4 Å (d) 8 Å, respectively

ACKNOWLEGEMENT

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

6. REFERENCES

[1] Bajaj, C. L., Lee, H. Y., Merkert, R. and Pascucci, V., NURBS based B-rep models for macromolecules and their properties, *4th Symposium on Solid Modeling and Applications*, May, 1997, pp 217-228.

[2] Bajaj, C. L., Lee, Dynamic maintenance and visualization of molecular surfaces, *Discrete Applied Mathematics*, Vol. 127, 2003, pp 23-51.

[3] Cheng, H.-L., Dey, T. K., Edelsbrunner, H. and Sullivan, J., Dynamic Skin Triangulation, *Discrete Computational Geometry* Vol. 25, 2001, pp525-568.

[4] Connolly, M. L., Analytical molecular surface calculation, *Journal of Applied Crystallography*, 16, 1983, pp548-558.

[5] Connolly, M, L., Molecular surface triangulation, *Journal of Applied Crystallography*, 18, 1985, pp499-505.

[6] Connolly, M, L., Molecular Surfaces: A Review, *Network Sci*. 1996.

[7] Connolly, M, L., Solvent-accessible surfaces of proteins and nucleic acids, *Science*, 221, 1983, pp 709-713.
[8] Edelsbrunner, H., Deformable smooth surface design, *Discrete Computational Geometry* Vol. 21, 1999, pp87-115.
[9] Gavrilova, M., Rokne, J., Updating the topology of the dynamic Voronoi diagram for spheres in Euclidean d-dimensional space. *Computer Aided Geometric Design* Vol. 20 No. 4, 2003, pp231–242.
[10] Kim, D.-S., Cho, Y. and Kim, D., Edge-tracing algorithm for Euclidean Voronoi diagram of 3D spheres, Proc. 16th Canadian Conference on Computational Geometry, 2004, pp 176–179.
[11] Kim, D.-S., Cho, Y., Kim, D., Kim, S., Bhak, J. and Lee, S.-H., Euclidean Voronoi diagram of 3D spheres and applications to protein structure analysis, *International Symposium on Voronoi Diagrams in Science and Engineering*, Sep., 2004, pp 137-144.
[12] Lee, B. and Richards, F.M., The interpretation of protein structures: estimation of static accessibility, *Journal of Molecular Biology*, Vol. 55, 1971, pp 379-400.
[13] Lozano-Perez, T., Spatial planning: a configuration space approach, *IEEE Transactions on Computers*, Vol. C-32, No. 2, 1983, pp 108-120.
[14] Maekawa, T., An overview of offset curves and surfaces, *Computer Aided Design* Vol. 31, 1999, pp 165-173.
[15] O'Donnell, M. L., The scientific and artistic uses of molecular surfaces, *Network Sci*. 1996.
[16] Okabe, A., Boots, B., Sugihara, K., Chiu, S.N., *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, 2nd ed., John Wiley & Sons, 1999.
[17] Piegl, L. and Tiller, W., *The NURBS Book*, Springer-Verlag, Berlin, 1995.
[18] RCSB Protein Data Bank. http://www.rcsb.org/pdb/, 2004.
[19] Richards, F. M., Areas, volumes, packing and protein structure, *Annu. Rev. Biophys. Bioeng.* Vol. 6, 1977, pp 151-176.
[20] Seidl, T. and Kriegel, H.-P., A 3D Molecular Surface Representation Supporting Neighborhood Queries, *3rd Conf. on Intelligent Systems for Molecular Biology* (*ISMB'95*) on, July, 1995, pp 350-358.
[21] Totorov, M. and Abagyan, R., The Contour-Buildup Algorithm to Calculate the Analytical Molecular Surface, *Journal of Structural Biology*, Vol. 116, 1996, pp 138-143.
[22] Varshney, A., Brooks, F. P., Jr. and Wright, W. V., Computing Smooth Molecular Surfaces, *IEEE Computer Graphics and Applications*, Vol. 14, No. 5, 1994, pp 19-25.
[23] Will, H.-M., *Computation of Additively Weighted Voronoi Cells for Applications in Molecular Biology*, Ph.D. Dissertation, Swiss Federal Institute of Technology, Zurich, 1999.