ELSEVIER

Discrete Applied Mathematics 71 (1996) 5–22

DISCRETE APPLIED MATHEMATICS

# Triangulating the surface of a molecule ☆

Nataraj Akkiraju [a,*], Herbert Edelsbrunner [a,b]

[a] *Department of Computer Science University of Illinois at Urbana-Champaign, Ill, USA*
[b] *Department of Computer Science, Hong Kong University Sci. Techn., Kowloon, Hong Kong*

Received 10 June 1995; revised 13 May 1996; accepted 20 May 1996

**Abstract**

Questions of chemical reactivity can often be cast as questions of molecular geometry. Common geometric models for proteins and other molecules are the space-filling diagram, the solvent accessible surface and the molecular surface. In this paper we present a new approach to triangulating the surface of a molecule under the three models, which is fast, robust, and results in topologically correct triangulations. Our computations are based on a simplicial complex dual to the molecule models. All proposed algorithms are parallelizable.

## 1. Introduction

One of the paradigms produced by the last quarter century of intense research on molecular biology is the relevance of geometric and topological reasoning for studying molecular phenomena. Shapes and topological structures of proteins play an important role in their functions. Accurate description of the shapes in terms of surfaces and volumes of proteins have led to rational approaches to the study of protein folding, protein–protein and protein–ligand interactions. Proteins are modeled geometrically, and questions of chemical reactivity are recast as questions of connectivity and fit [12, 13]. Possibly the most popular geometric representation invoked to rationalize the behavior of proteins is the space filling model. Sample applications include the attempt to link the driving force for protein folding to solvent accessible surface area, and the reduction of affinity for ligand protein binding to the mechanical fit of a ligand with a protein receptor site.

*Models for molecules.* Common geometric representations for proteins and other molecules are the space-filling, the solvent-accessible, and the molecular surface models. The *space filling* or *SF model*, introduced by Lee and Richards [12], interprets a protein as the union of possibly overlapping spherical balls in $\mathbb{R}^3$. Each ball represents

☆ The research of both authors is partially supported by the Office of Naval Research. Herbert Edelsbrunner is also supported through the Alan T. Waterman award, grant CCR-9118874.

* Corresponding author. E-mail: nataraj-akkiraju@mentorg.com.


*PII* S0166-218X(96)00054-6

an atom and its size is determined by the van der Waals radius. The *solvent accessible* or *SA model*, was introduced to study the interaction between a protein and a solvent modeled as a spherical ball [13]. The solvent is deflated to a point and the spherical balls representing atoms in the protein are inflated by the solvent radius. Geometrically, there is little difference between the SF and SA models: both are unions of spherical balls albeit of different sizes. The *molecular surface* or *MS model* is obtained by rolling the sphere representing the solvent over the SF model [13].

*Surface triangulations.* The surface of a molecule is useful in studying structure of and interaction between proteins and other molecules. A particularly important problem in this context is the protein–substrate docking problem. A topologically correct triangulation of a protein surface has several applications: packing defects in proteins may be identified, water molecules may be located relative to the protein, shape and location of cavities can be computed, mathematical functions defined on the surface may be contoured, and local optima of such functions may be identified. Visual inspection of three-dimensional molecular surfaces is useful in identifying binding sites and in drug design where molecules are designed to fit a receptor or active site. Representation of the surface by a set of triangles is faster and more efficient than other representation schemes. Previous work on surface triangulations including [4, 14, 15] is rather ad hoc in nature, some of the methods are numerically instable, and the resulting triangulations are not always topologically correct. Connolly [4] describes a scheme where the triangulation is computed from a curved surface made up of pieces of spheres and tori that join at arcs of circles. The curved surface is calculated from atomic coordinates by means of an analytical molecular surface algorithm [3]. Each face of the curved surface is defined by a set of cycles of edges that form its boundary. The basic idea behind the triangulation algorithm is recursive subdivision. Each face is divided into smaller faces recursively till all the faces have been replaced by a collection of triangles, each smaller than a specified size. Varshney et al. [14] present a parallel algorithm for computing molecular surfaces in time $O(k \log k)$ over $n$ processors, where $k$ is the average number of neighbors of an atom and $n$ is the number of atoms. They compute intersection patterns for each atom in parallel and from this information generate a tesselation of the molecular surface. Zauhar and Morgan [15] also start with a description of the curved surface which they generate from atomic coordinates. Next, they generate points on each face and triangulate the face in an incremental fashion starting from a boundary edge.

In this paper we present a different approach to computing a triangulation of the surface of a molecule under the SF, the SA, and the MS models. Our computations are based on a simplicial complex dual to the sphere model of the molecule, see [10]. The use of a simulated perturbation scheme eliminates all degeneracies and allows us to assume the input is in general position. The geometric integrity of this complex guarantees the topological correctness of the surface triangulation. We also make use of a new approach to tesselating sphere patches which makes our triangulation particularly suitable for visualization.

*Outline.* Section 2 introduces basic geometric concepts, including a simplicial complex dual to the above molecule models. Section 3 describes the details of the

relationship between the models and the complex. Section 4 shows how to construct a triangulation of the boundary of an SF or SA model. Section 5 extends these results to MS models. Section 6 discusses the sequential and parallel implementations and addresses robustness problems. Section 7 concludes the paper with a brief discussion of its main contributions.

## 2. Spheres and simplices

This section formally defines the dual complex of a union of spherical balls in three-dimensional real space $\mathbb{R}^3$. An example of such a union of balls is the space filling model of a molecule: each atom is represented by a ball whose size is determined by its van der Waals radius. Different radii are needed for different types of atoms.

*Basic definitions.* $|yz|$ is the *Euclidean distance* between two points $y, z \in \mathbb{R}^3$. A (*spherical*) *ball* or 3-*ball* in $\mathbb{R}^3$ is a set of the form

$$b = \{x \in \mathbb{R}^3 \mid |xz| \leqslant \varrho\};$$

$z \in \mathbb{R}^3$ is the *center* and $\varrho > 0$ is the *radius* of $b$. 2-*balls*, 1-*balls*, and 0-*balls* are obtained by intersecting a 3-ball with planes, lines, and points, respectively, containing its center. A $(k-1)$-*sphere* is the boundary of a $k$-ball. It follows every 2-ball is a disk, every 1-ball is a line segment, every 0-ball is a point, every 2-sphere is a sphere, every 1-sphere is a circle, and every 0-sphere is a pair of points.

A $k$-*simplex*, $\sigma$, is the convex hull of $k+1$ affinely independent points. The *dimension* of $\sigma$ is $\dim \sigma = k$. In $\mathbb{R}^3$, at most 4 points can be affinely independent, so we only have 3-, 2-, 1-, and 0-simplices referred to as *tetrahedra*, *triangles*, *edges*, and *vertices*, respectively. An $\ell$-*face* of $\sigma$ is a simplex defined by $\ell + 1 \leqslant k + 1$ vertices of $\sigma$.

*Simplicial complexes.* An *abstract simplicial complex* is a finite system $\mathscr{A}$ of sets so $X \in \mathscr{A}$ and $Y \subseteq X$ implies $Y \in \mathscr{A}$. Since $\emptyset$ is subset of every set it is an element of every abstract simplicial complex. Define $\operatorname{card} X$ to be the cardinality of the set $X$. The *dimension* of $\mathscr{A}$ is $\dim \mathscr{A} = \max_{X \in \mathscr{A}} \{\operatorname{card} X - 1\}$ and the *vertex set* of $\mathscr{A}$ is $\operatorname{Vert} \mathscr{A} = \cup \mathscr{A}$. A 3-dimensional geometric picture of $\mathscr{A}$ can be obtained through a map $\varepsilon : \operatorname{Vert} \mathscr{A} \to \mathbb{R}^3$: each element is mapped to a point and each set $X \in \mathscr{A}$ is mapped to the convex hull of the points in $\varepsilon(X)$. The set of simplices, $\{\operatorname{conv} \varepsilon(X) \mid X \in \mathscr{A}\}$, is a *geometric realization* of $\mathscr{A}$ if $X, Y \in \mathscr{A}$ implies $\operatorname{conv} \varepsilon(X) \cap \operatorname{conv} \varepsilon(Y) = \operatorname{conv} \varepsilon(X \cap Y)$. A (*geometric*) *simplicial complex*, $\mathscr{K}$, is the geometric realization of an abstract simplicial complex. Its *dimension* is the same as the dimension of the corresponding abstract simplicial complex. The *underlying space* of $\mathscr{K}$ is $|\mathscr{K}| = \cup_{\sigma \in \mathscr{K}} \sigma$. A *subcomplex* of $\mathscr{K}$ is a simplicial complex $\mathscr{L} \subseteq \mathscr{K}$.

Large abstract and geometric simplicial complexes can be generated using the nerve operation. The *nerve* of a collection $S$ of sets is

$$\operatorname{Nrv} S = \{X \subseteq S \mid \bigcap X \neq \emptyset\}.$$

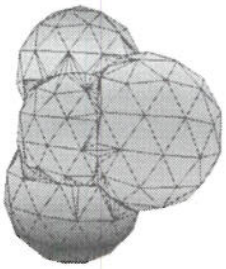

Fig. 1. A surface triangulation for the union of 4 spherical balls. It consists of 4 patches separated by 6 arcs and 4 corners.

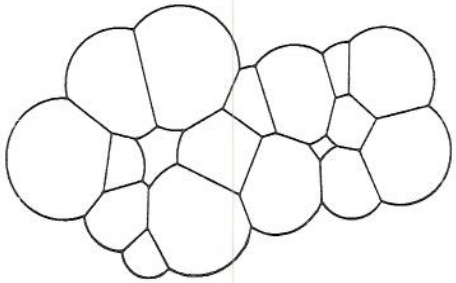

Fig. 2. Decomposition of a union of disks showing the cells $U_b$.

The nerve is an abstract simplicial complex because $X \in \mathrm{Nrv}\, S$ and $Y \subseteq X$ implies $Y \in \mathrm{Nrv}\, S$.

Let $B$ be a finite set of 3-balls in $\mathbb{R}^3$. A *triangulation* of the boundary of $\cup B$ is a two-dimensional simplicial complex $\mathscr{T}$ whose underlying space is homeomorphic to the boundary, $\mathrm{Sur}\, B = \mathrm{bd}\,(\cup B)$, see Fig. 1. Our interest is in computing $\mathscr{T}$. The data structure representing $\mathscr{T}$ stores triangles and pointers between adjacent triangles.

*Voronoi decomposition.* We assume the balls in $B$ are in general position. Among other things, this implies the centers of every $k+1 \leqslant 4$ 3-balls are affinely independent and thus define a proper $k$-simplex. An algorithmic justification for this assumption can be found in [9]. Consider a 3-ball $b$ with center $z \in \mathbb{R}^3$ and radius $\varrho$, and a point $x \in \mathbb{R}^3$. The *weighted distance* of $x$ from $b$ is $\pi_b(x) = |xz|^2 - \varrho^2$. The (*weighted*) *Voronoi cell* of $b \in B$ is

$$V_b = \{x \in \mathbb{R}^3 \mid \pi_b(x) \leqslant \pi_c(x), \forall c \in B\}.$$

$\mathsf{V} = \{V_b \mid b \in B\}$ is the set of Voronoi cells. The Voronoi cells decompose the union of balls, $\cup B$, into cells with disjoint interiors. Some of the cells share parts of their boundary, which will be essential in defining a simplicial complex reflecting the intersection pattern. For each $b \in B$ consider the intersection of its Voronoi cell with $\cup B$, $U_b = V_b \cap \cup B$, and note that $U_b = V_b \cap b$. Since both $V_b$ and $b$ are convex it follows $U_b$ is also convex, see Fig. 2.

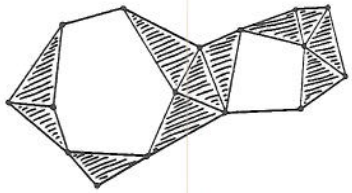

Fig. 3. The dual complex of the disk union in Fig. 2.

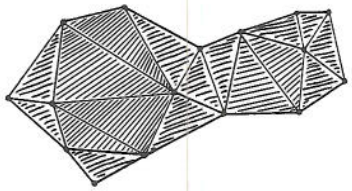

Fig. 4. The Delaunay complex of the disk union in Fig. 2.

*Dual complex.* The simplicial complex mentioned in the abstract and the introduction is constructed by taking the nerve of the above decomposition of $\cup B$. Let $\mathsf{U} = \{U_b \mid b \in B\}$. The *dual complex* of $\cup B$ is

$$\operatorname{Cpx} B = \{\operatorname{conv} \varepsilon(X) \mid X \in \operatorname{Nrv} \mathsf{U}\},$$

where $\varepsilon(U_b)$ is the center of $b$, see Fig. 3. $\operatorname{Cpx} B$ is indeed a simplicial complex. To see this consider

$$\operatorname{Del} B = \{\operatorname{conv} \varepsilon(X) \mid X \in \operatorname{Nrv} \mathsf{V}\},$$

the (*weighted*) *Delaunay complex* of $B$, see e.g. [8]. It is known and not very difficult to prove that the simplices in $\operatorname{Del} B$ intersect properly and thus form a simplicial complex. $\operatorname{Cpx} B$ is a subset of $\operatorname{Del} B$, see Fig. 4, which implies it is also a simplicial complex.

The *dual shape* of $\cup B$ is the underlying space of the dual complex, $|\operatorname{Cpx} B|$. A *face* of $|\operatorname{Cpx} B|$ is a simplex $\sigma \in \operatorname{Cpx} B$ that belongs to the boundary of $|\operatorname{Cpx} B|$. An important result proved in [8] is the following.

(1) $\cup B$ and $|\operatorname{Cpx} B|$ are homotopy equivalent, and more specifically, $|\operatorname{Cpx} B|$ is a deformation retract of $\cup B$.

As a consequence, $|\operatorname{Cpx} B| \subseteq \cup B$ and every cavity of $\cup B$ is contained in the corresponding cavity of $|\operatorname{Cpx} B|$. This turns out to be useful in the detection and construction of cavities in proteins.

## 3. Links and patches

There is a close relationship between the face structure on the surface of the ball union and the face structure of the dual shape. This section describes this relationship in appropriate detail.

*Faces and face structure.* The boundary of $\cup B$ consists of spheres of various dimensions. Denote the sphere bounding a ball $b$ by $\bar{b} = \mathrm{bd}(b)$ and the clipped sphere by $\check{b} = V_b \cap \bar{b}$. $\check{b}$ is the spherical part of the boundary of the convex cell $U_b \in \mathsf{U}$. The components of $\check{b}$ are 2-*faces* of $\cup B$. Lower-dimensional faces can be defined by considering pair and triple intersections of clipped spheres. By general position assumption, the intersection of 2 clipped spheres is either empty or 1-dimensional, and the intersection of 3 clipped spheres is either empty or consists of 1 or 2 points. The components of the intersection are 1-*faces* and 0-*faces* of $\cup B$. We refer to a 2-, 1-, 0-face of $\cup B$ as a *patch*, an *arc*, a *corner*. Each patch is bounded by one or more cycles of arcs, unless it is an entire sphere and without boundary. Each arc is bounded by 2 corners, unless its is an entire circle and without boundary. The following result can be found in [8].

(2) A simplex $\sigma = \mathrm{conv}\,\varepsilon(X)$, $X \subseteq \mathsf{U}$, is a face of $|\mathrm{Cpx}\,B|$ iff the intersection of the corresponding clipped spheres is non-empty:

$$\bigcap_{U_b \in X} \check{b} \neq \emptyset.$$

In other words, if we know the faces of $|\mathrm{Cpx}\,B|$ we also know which 2-, 1-, and 0-spheres defined by balls in $B$ contribute faces to the surface, $\mathrm{Sur}\,B = \mathrm{bd}(\cup B)$. The faces themselves, and their incidence structure, are determined by certain subcomplexes of $\mathrm{Cpx}\,B$. This is explained in the next two paragraphs.

*Joins, stars, and links.* We denote the edge connecting two points $p, q \in \mathbb{R}^3$ by $pq$. The *join* of two sets $P, Q \subseteq \mathbb{R}^3$ is

$$P * Q = \bigcup_{p \in P, q \in Q} pq.$$

It is defined if $P \cap Q = \emptyset$ and any two edges $pq$ and $p'q'$, with $p, p' \in P$ and $q, q' \in Q$, are either disjoint or meet only at a common endpoint. By definition, $P * \emptyset = \emptyset * P = P$. The *star* of a simplex $\sigma \in \mathscr{K}$ is $\mathrm{St}\,\sigma = \{\tau \in \mathscr{K} \mid \sigma \subseteq \tau\}$. The *link* of $\sigma$ is

$$\mathrm{Lk}\,\sigma = \{\tau \in \mathscr{K} \mid \sigma * \tau \in \mathrm{St}\,\sigma\}.$$

*Surface contributions.* Consider a 3-ball, $b \in B$, its bounding 2-sphere, $\bar{b}$, its contribution to the surface, $\check{b}$, and the part buried inside the ball union,

$$\mathrm{cl}\,(\bar{b} - \check{b}) = \bar{b} \cap \bigcup (B - \{b\}).$$

The buried part of $\bar{b}$ is the union of finitely many circular caps on a 2-sphere, namely the union of all caps in $C = \{\bar{b} \cap c \mid c \in B - \{b\}\}$. $\cup C$ is similar to $\cup B$, only one dimension lower and on a 2-sphere rather than in a plane. It is therefore not surprising there is a simplicial complex, $\mathscr{L}$, dual to $\cup C$. This complex is the link of $u = \varepsilon(U_b)$, which is a vertex in $\mathrm{Cpx}\,B$. $\mathscr{L}$ is two-dimensional and it is natural and appropriate to define its boundary relative to a 2-dimensional surface containing $\mathscr{L}$. In general, the underlying space of the link of $u$ in $\mathrm{Del}\,B$ is such a surface.

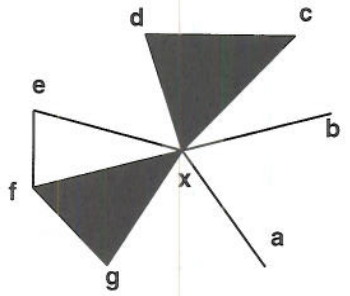


Fig. 5. The link of vertex $x$ is $\{\emptyset, a, b, c, d, e, f, g, cd, fg\}$. The link of edge $xc$ is $\{\emptyset, d\}$.

Only if $u$ lies on the boundary of $|\mathrm{Del}\, B| = \mathrm{conv}\, \varepsilon(\mathsf{U})$ we need to artificially add simplices to complete $\mathrm{Del}\, B$ to a triangulation of the 3-sphere. Assuming such a completion, a *face* of $|\mathscr{L}|$ is a simplex in $\mathscr{L}$ that belongs to the relative boundary of $|\mathscr{L}|$. The following results similar to (1) and (2) can be found in [8].

(3) $\cup C$ and $|\mathscr{L}|$ are homotopy equivalent.

(4) A simplex $\sigma = \mathrm{conv}\, \varepsilon(X)$, $U_b \notin X \subseteq \mathsf{U}$, is a face of $|\mathscr{L}|$ iff the intersection of the corresponding clipped circles is non-empty:

$$\bigcap_{U_c \in X} (\check{c} \cap \bar{b}) = \bar{b} \cap \bigcap_{U_c \in X} \check{c} \neq \emptyset.$$

Results (3) and (4) can be extended to the boundary contribution of a 1-sphere and the link of an edge, and to the boundary contribution of a 0-sphere and the link of a triangle. Consider two 3-balls, $b, d \in B$, the 1-sphere $\bar{bd} = \bar{b} \cap \bar{d}$, its contribution to the surface, $\check{bd}$, and the part buried inside the ball union,

$$\mathrm{cl}\,(\bar{bd} - \check{bd}) = \bar{bd} \cap \bigcup (B - \{b, d\}).$$

The buried part of $\bar{bd}$ is the union of finitely many circular arcs on a 1-sphere, namely the union of all arcs in $F = \{\bar{bd} \cap c \mid c \in B - \{b, d\}\}$. The simplicial complex, $\mathscr{L}$, dual to $\cup F$ is the link of the edge $uv$, $u = \varepsilon(U_b)$ and $v = \varepsilon(U_d)$. We can similarly relate the boundary contribution of a 0-sphere to the link of a triangle. Section 4 presents the algorithm recovering the patches, arcs, and corners of $\cup B$ from the links of vertices, edges, and triangles in $\mathrm{Cpx}\, B$.

## 4. SF and SA models

This section discusses the surface triangulation algorithm for the union of a set of balls in further detail. It is instructive to first study the 2-dimensional case, where we consider the union of disks in $\mathbb{R}^2$. The problem in $\mathbb{R}^2$ is similar to tracing the boundary of the surface contribution of a single 2-sphere in $\mathbb{R}^3$.

*Union of disks.* Let $D$ be a finite set of 2-balls or disks in $\mathbb{R}^2$, see Fig. 2. The union, $\cup D$, consists of one or more components, and each component is bounded by one or

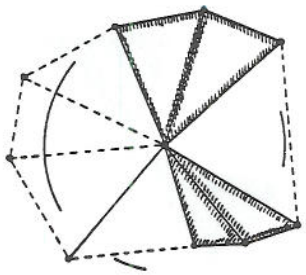

Fig. 6. The boundary contribution of a circle and the corresponding star and link. Non-shaded triangles and broken edges indicate simplices of Del $D$ that do not belong to the dual complex.

more cycles. More globally, the boundary of the union, $\operatorname{Sur} D = \operatorname{bd}(\cup D)$, consists of finitely many cycles, and each cycle is a cyclic sequence of circular arcs. A cycle can consist of as few as one arc, in which case the arc is a complete circle. If there are two or more arcs then each arc is bounded by 2 corners, which it shares with the adjacent arcs in the cycle.

The two-dimensional specialization of the surface triangulation problem in $\mathbb{R}^3$ asks for approximating each arc by a chain of line segments. A *triangulation* of $\operatorname{Sur} D$ is a one-dimensional simplicial complex whose underlying space is homeomorphic to $\operatorname{Sur} D$. It suffices to construct a chain per arc and to connect the chains at their begin and end points to form appropriate cycles.

Consider a disk, $d \in D$, its bounding circle, $\bar{d} = \operatorname{bd}(d)$, its contribution to the boundary, $\breve{d} = V_d \cap \bar{d}$, and the part buried inside the union,

$$\operatorname{cl}(\bar{d} - \breve{d}) = \bar{d} \cap \bigcup (D - \{d\}).$$

The buried part is the union of the one-dimensional caps in $F = \{\bar{d} \cap e \mid e \in D - \{d\}\}$. It is possible a disk $e$ completely contains $d$, in which case $\bar{d} \cap e = \bar{d}$ and $\breve{d} = \emptyset$. Symmetrically, if $d$ contains $e$ then $\bar{d} \cap e = \emptyset$. In all other cases, a cap $\bar{d} \cap e$ is bounded by the two intersection points of $\bar{d}$ and $\bar{e}$.

*Traversing a link.* We can trace the components of $\breve{d}$ by traversing the link $\mathscr{L} = \operatorname{Lk} u$ of the vertex $u = \varepsilon(V_d)$ in Del $D$, see Fig. 6. For convenience assume Del $D$ is completed to a triangulation of a 2-sphere so we do not need to distinguish between the general case, when $u$ belongs to the interior of $|\operatorname{Del} D|$, and the other case, when it belongs to the boundary of $|\operatorname{Del} D|$. With this assumption, $\mathscr{L}$ is always a cycle of vertices and edges. For a vertex $v \in \mathscr{L}$, let $\mathsf{next}(v)$ be the successor vertex in a counterclockwise order about $u$. Let $\mathsf{left}(u, v)$ be the triangle in Del $D$ to the left of the edge $uv$ directed from $u$ to $v$. We assume the data structure for Del $D$ supports functions returning the next vertex in a link and the triangle to the left of a directed edge in constant time. The following algorithm computes polygonal chains approximating the contribution of $\bar{d}$ to $\operatorname{Sur} D$. It assumes $\breve{d} \neq \emptyset$, which is the case iff $u$ is a vertex of $|\operatorname{Cpx} D|$. Let $v_0$ be an arbitrary vertex of $\mathscr{L}$ that is also a vertex of $|\operatorname{Cpx} D|$. Assume also that the edge connecting $v_0$ with $v_1 = \mathsf{next}(v_0)$ does not belong to $\mathscr{L}$, or equivalently,

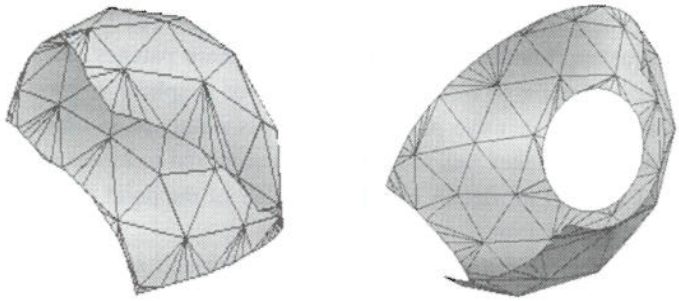

Fig. 7. Examples of patches. The second patch has a hole.

$\mathsf{left}(v_0v_1) \notin \operatorname{Cpx} D$.

```
a := v0;
loop
  while left(u,a) ∈ Cpx D do a := next(a) endwhile;
  b := next(a);
  while b ∉ Cpx D do b := next(b) endwhile;
  chain(u,a,b);
  if a = v0 or b = v0 then exit endif;
  a := b;
forever.
```

$\mathsf{chain}(u,a,b)$ computes a polygonal approximation to the arc on circle $\bar{d}$ that begins at the counterclockwise second intersection point of $\bar{d}$ and $\bar{a}$ and ends at the counterclockwise first intersection point of $\bar{d}$ and $\bar{b}$. In the first `while` loop, if the triangle to the left of $ua$ belongs to $\operatorname{Cpx} D$ we advance $a$, which is equivalent to erasing another buried piece of $\bar{d}$. In the second `while` loop, if $b$ is not in the dual complex of $\cup D$ we advance $b$, which is equivalent to adding another piece of $\bar{d}$ to $\check{d}$.

*Union of balls.* We return to three dimensions. $B$ is a finite set of 3-balls in $\mathbb{R}^3$, as usual. The surface, $\operatorname{Sur} B = \operatorname{bd}(\cup B)$, is the union of patches, and to triangulate $\operatorname{Sur} B$ we approximate each patch by a homeomorphic two-dimensional simplicial complex. This amounts to selecting points on each patch and connecting them with edges and triangles to form a connected surface piece with boundary, see Fig. 7. Note the patch is not necessarily simply connected. After triangulating all patches we glue them together at matching boundary arcs.

Let $b \in B$ be a 3-ball and consider $\bar{b}$, $\check{b}$, and $\cup C$ as in section 3. It was mentioned earlier that $\check{b}$, the contribution of $\bar{b}$ to $\operatorname{Sur} B$, is the complement of a union of circular caps, $\cup C$. Its boundary consists of one or more cycles of circular arcs, and these cycles are computed and approximated as described in the two-dimensional discussion above using the link of a vertex in lieu of the dual two-dimensional complex. We shall refer to the edges approximating the circular arcs as *arc edges* and their end points as *arc points*.

Details of a technique for generating points on a sphere can be found in [1], where convex polyhedral approximations of spheres and sphere patches are studied. Starting with a random distribution of a finite set of points on a sphere, a good approximation is constructed during an iterative process that moves points in an attempt to maximize the surface area of the convex hull. This scheme is referred to as the area maximization heuristic (amh). See [1] for details about the quality of the point distribution generated by amh and the computational time.

We precompute and store a point distribution on the unit sphere using amh. These points are translated so they now lie on $\bar{b}$. The arc points are added to this point set. Next, a (possibly non-convex) polytope $P$ is constructed on this point set. The attempt is to get a convex polytope but if convexity contradicts that all arc edges belong to the boundary, then convexity is sacrificed. $P$ has the property that all arc edges appear on the boundary and $P$ exhibits non-convexity only in the vicinity of the arc edges. $P$ is referred to as the *constrained* convex hull, see [1].

Once we have constructed $P$, we can erase parts of it not needed in the approximation of $\check{b}$. The parts to be erased are bounded by cycles of arc edges. We start a depth first traversal with a triangle in $P$ that we know is used in the approximation of $\check{b}$ and hence needs to be reported. During the search, whenever we cross an arc edge, we either leave or enter $\check{b}$. We keep track of this parity information to appropriately accept and reject triangles of $P$.

## 5. MS models

The molecular surface or MS model defined by Richards [13] modifies the SF model using the concept of a rolling sphere representing the solvent. Intuitively, cusps and crevices are filled up to the extent they cannot be reached by the rolling sphere touching but not overlapping the SF model. We first define the resulting surface and then show how the triangulation of a corresponding SA surface can be modified to give a triangulation of the molecular surface.

*Rolling sphere definition.* Let $\cup B$ be the SF model of the considered molecule, and let $R$ be the rolling sphere representing the solvent. Assuming general position, $R$ can touch $\operatorname{Sur} B$ in only 1, 2, or 3 points, in the second case the 2 points cannot be diametrically opposite, and in the third case the 3 points cannot lie on a great-circle of $R$. The center $p$ of $R$ is used to refer to a particular position $R_p$ of the rolling sphere. Only positions where the rolling sphere touches $\operatorname{Sur} B$ but does not overlap $\cup B$ are considered.

An important definition is the *geodesic hull* of the contact points, denoted $\operatorname{gd} p$. If $R_p$ touches $\operatorname{Sur} B$ in only 1 point, $x$, then $\operatorname{gd} p = x$. If $R_p$ touches $\operatorname{Sur} B$ in 2 points, $x$ and $y$, then $\operatorname{gd} p$ is the geodesic on $R$ connecting $x$ and $y$; it is the shorter of the two connecting great-circle arcs. If $R_p$ touches $\operatorname{Sur} B$ in 3 points, $x, y, z$, then $\operatorname{gd} p$ is the smaller of the two spherical triangles on $R$ bounded by the geodesics connecting $x$ and $y$, $y$ and $z$, and $z$ and $x$. The *molecular surface*, $\operatorname{Mol} B$, is the union of all geodesic hulls over all positions of $R$ touching $\operatorname{Sur} B$.

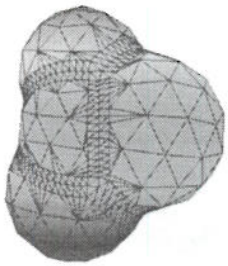

Fig. 8. The molecular surface of the union of same 4 balls as in Fig. 1. It consists of 4 convex patches separated by 6 saddles and 4 concave patches.

Following Connolly [3, 4] we differentiate regions on $\operatorname{Mol} B$ by the number of contact points generating geodesic hulls, see Fig. 8. The *contact region* is the union of $\operatorname{gd} p$ over all positions with only 1 contact point. The *reentrant region* is the union of $\operatorname{gd} p$ over all positions with 2 or 3 contact points. The components of the contact region are referred to as *convex patches*. The reentrant region consists of *saddles*, that are components of the union of geodesics connecting pairs of contact points, and of *concave patches*, that are spherical triangles on the rolling sphere generated by triple contacts. The convex patches, saddles, and concave patches meet at their bounding arcs forming the molecular surface.

*Self-intersections.* The above definition of $\operatorname{Mol} B$ as the union of geodesics allows for possible self-intersections. Indeed, already 2 balls generate a self-intersecting molecular surface if the gap between them is only slightly smaller than the diameter of the rolling sphere.

In the molecular surface literature, these self-intersections are commonly referred to as degeneracies, and they are removed by clipping $\operatorname{Mol} B$ at the curves of self-intersection [4]. We take a different viewpoint tolerating self-intersections: $\operatorname{Mol} B$ is a genuine surface, only its embedding in $\mathbb{R}^3$ is not intersection-free. This interpretation has the advantage of a direct correspondence between the patches of $\operatorname{Mol} B$ and the faces of the SA model defined by the solvent sphere, $R$.

*Correspondence to SA surface.* Recall the definition of $\operatorname{Mol} B$ as the union of geodesic hulls. Now, instead of taking the union of the $\operatorname{gd} p$ take the union of the points $p$. The result is the surface of another ball union, $\cup B_\alpha$. $\alpha$ is the radius of the rolling sphere, and for each ball $b$ with radius $\varrho$ in $B$ the concentric ball $b_\alpha$ with radius $\varrho + \alpha$ belongs to $B_\alpha$.

Each convex patch of $\operatorname{Mol} B$ corresponds to a patch of $\operatorname{Sur} B_\alpha$. To be specific, consider a ball $b_\alpha \in B_\alpha$ and its contribution $\check{b}_\alpha$ to $\operatorname{Sur} B_\alpha$. The components of $\check{b}_\alpha$ are patches of $\operatorname{Sur} B_\alpha$. The convex patches of $\operatorname{Mol} B$ on $\bar{b}$ are obtained by shrinking $\check{b}_\alpha$ towards the shared center until the radius is the same as that of $b$. More globally, the contact surface of $\operatorname{Mol} B$ is obtained by simultaneously shrinking all patches of $\operatorname{Sur} B$, each patch towards the center of its own sphere, see Fig. 9.

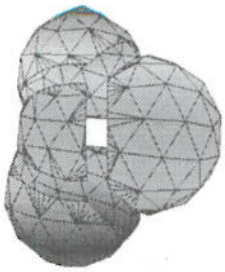

Fig. 9. Gaps between (convex) patches opening up during the shrinking process.

Each saddle of $\operatorname{Mol} B$ corresponds to an arc of $\operatorname{Sur} B_\alpha$. Suppose the arc belongs to $\bar{b}_\alpha \cap \bar{c}_\alpha$. A copy of the arc shrinks towards the center of $b$, and another copy shrinks towards the center of $c$. The two shrunk copies form two boundary arcs of the saddle bridging the opening gap, see Fig. 9. These two boundary arcs connect the saddle to convex patches on $\bar{b}$ and on $\bar{c}$. The other two boundary arcs connect the saddle to concave patches corresponding to the corners limiting the arc.

Let $p$ be a corner of $\operatorname{Sur} B_\alpha$, and let $d \in B$ so $p \in \bar{b}_\alpha \cap \bar{c}_\alpha \cap \bar{d}_\alpha$. A copy of $p$ shrinks towards the center of $b$, another towards the center of $c$, and a third towards the center of $d$. At the end of the shrinking process, the three copies coincide with the contact points $x, y, z$ of $R$ at position $p$, see figure 9. The geodesic hull of $x, y, z$ is the concave patch of $\operatorname{Mol} B$ that corresponds to the corner $p$. The 3 boundary arcs connect the concave patch to 3 saddles, one for each arc of $\operatorname{Sur} B_\alpha$ ending at $p$.

*Triangulating the molecular surface.* The correspondence between $\operatorname{Mol} B$ and $\operatorname{Sur} B_\alpha$ suggests we derive a triangulation of the former from the triangulation of the latter.

`Step 1`: Triangulate $\operatorname{Sur} B_\alpha$.

`Step 2`: Shrink the triangulated patches of $\operatorname{Sur} B_\alpha$ towards their respective centers.

`Step 3`: Triangulate the saddles and the concave patches used to bridge the opening gaps.

`Step 1` is explained in Section 4, and `step 2` is straightforward. `Step 3` needs some clarification. Each saddle is the homeomorphic image of a rectangle in $\mathbb{R}^2$. It is easy to triangulate it choosing a $k$-by-$\ell$ regular grid of points. Edges are drawn in three directions: vertical, horizontal, and diagonal. Note that the adjacent convex patches are already triangulated, which implies $\ell$ and the $\ell + \ell$ points along two opposite sides of the rectangle are prescribed. Each of the other two sides carries $k$ points, and $k$ can be chosen suitable for the length of these sides, or rather the length of their homeomorphic images. The vertices of the rectangle triangulation are mapped to points on the saddle using the homeomorphism mentioned before, and the edges and triangles are added connecting these points. The result is a triangulating approximation of the saddle, see Fig. 10.

Finally, the concave patches need to be triangulated. We use the same method as for the (convex) patches of $\operatorname{Sur} B_\alpha$, see [1]. The chains of line segments approximating the

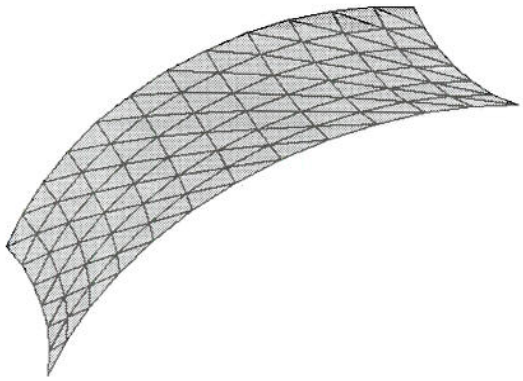

Fig. 10. Saddle surface.

3 boundary arcs of a concave patch are fixed by the triangulations of the 3 adjacent saddles.

## 6. Implementation Issues

The implementation of the algorithms described in sections 4 and 5 is by no means straightforward. Some of the steps require sophisticated data structures to achieve satisfactory performance, others pose challenging robustness questions. This section reviews the overall algorithm and discusses some of the more delicate implementation issues.

*Overall algorithm.* The triangulation of the molecular surface, $\mathrm{Mol}\,B$, is based on the triangulation of $\mathrm{Sur}\,B_\alpha$, which in turn relies on the availability of $\mathrm{Del}\,B_\alpha$ and $\mathrm{Cpx}\,B_\alpha$. We assume the input consists of a finite set $B$ of 3-balls in $\mathbb{R}^3$, each specified by its center and radius, and by the solvent radius $\alpha > 0$.

`Step C1`: Inflate the radius of each ball in $B$ by $\alpha$; the result is $B_\alpha$.

`Step C2`: Construct $\mathrm{Del}\,B_\alpha$.

`Step C3`: Construct $\mathrm{Cpx}\,B_\alpha$ by selecting the appropriate simplices in $\mathrm{Del}\,B_\alpha$.

`Step C1` is straightforward. `Step C2` is quite demanding, in particular if robustness and performance are stressed. We use the publically available Delaunay triangulation software discussed in [10, 2]. It is based on integer or real fixed-point coordinates and uses exact arithmetic to guarantee a geometrically and topologically consistent representation of $\mathrm{Del}\,B_\alpha$. The use of a simulated perturbation admits the assumption of general position and unwanted effects such as zero-volume tetrahedra are removed in a post-processing step. The output is a triangle-based data structure similar to the edge-facet data structure described in [7]. We refer to this as the *triangle-edge* data structure [10]. The atomic unit of the triangle-edge data structure is the triangle-edge pair $a = \langle \sigma, i \rangle$ with $0 \leqslant i \leqslant 5$. It identifies 6 versions of the triangle $\sigma$, one for each of its six directed edges. Each triangle has two edge rings associated with it, one traversing the edges in a clockwise order and the other in a counter clockwise order.

Similarly, each edge defines two triangle rings traversing the incident triangles in the two opposite orders. The triangle-edge data structure allows for traversing the link of an edge or a vertex in constant time per step. The algorithm used to construct the Delaunay complex takes time and storage $O(n^2)$ in the worst case where $n = |B|$. Experimental evidence suggests that it performs significantly better for most point sets, especially proteins and other molecules.

The software also includes a program that builds a data structure storing a sequence of Delaunay simplices that contains $\mathrm{Cpx}\, B_\alpha$ as a subcomplex of $\mathrm{Del}\, B_\alpha$. This program takes time $O(m \log m)$ where $m$ is the number of simplices in $\mathrm{Del}\, B_\alpha$, which is quadratic in the worst case and seemingly linear in most practical cases. We generate $\mathrm{Cpx}\, B_\alpha$ by reporting all simplices stored up to a certain index in the sequence. For further details on how this sequence and the index are computed, see [10].

`Step S1`: Traverse links of edges in $\mathrm{Del}\, B_\alpha$ to compute and connect approximations to arcs of $\mathrm{Sur}\, B_\alpha$.

`Step S2`: Traverse links of vertices in $\mathrm{Del}\, B_\alpha$ to compute and connect triangulating approximations to patches of $\mathrm{Sur}\, B_\alpha$.

The time taken for link traversal in steps `S1` and `S2` is bounded by the number of simplices in $\mathrm{Del}\, B_\alpha$. The time taken for generating approximations of arcs and patches depends on parameters controlling the density of the point distribution on arcs and spheres. The program provides the user the ability to specify these parameters at run time. If the distribution is very dense, then the patch computation phase will take longer and vice versa.

`Steps S1` and `S2` pose serious robustness problems discussed below. `Step S2` completes the triangulation of the solvent accessible model, $\cup B_\alpha$. The only change necessary to compute a triangulation of the space filling model, $\cup B$, is the omission of `step C1` and the substitution of $B$ for $B_\alpha$ in `steps C2, C3, S1,` and `S2`.

`Step M1`: Shrink the patches of $\mathrm{Sur}\, B_\alpha$ to obtain the convex patches of $\mathrm{Mol}\, B$.

`Step M2`: Compute triangulating approximations of all saddles and concave patches and add them to bridge the gaps between convex patches.

`Step M2` completes the triangulation of the molecular surface. Each component is an orientable surface represented by a triangle-based data structure similar to the quad-edge structure described in [11]. Each element in the triangulation stores the three end points of the triangle and three pointers to neighboring triangles. It admits local navigation in constant time per step through explicitly storing adjacency information among triangles. This information effectively glues the triangles along shared edges and thus triangulated patches along shared boundary arcs. Adjacencies are computed in constant time per triangle using radix sort on the set of edges. The time complexity of building the triangulation is linear in the size of the triangulation.

*Robustness.* Our software for patch computation does not perform any numerical computations in the sense of using floating-point numbers and arithmetic. Still, we run into similar robustness problems resulting from inconsistency between different representations: the Delaunay complex, $\mathrm{Del}\, B_\alpha$, and the triangulated surface patches. To explain we need more details about the triangulation algorithm in `steps S1` and `S2`.

To approximate a circular arc in `step S1` we pick points on the arc, or rather near the arc because a fixed number of bits per coordinate may allow only a very small number of points exactly on the arc, possibly no point at all. The same applies to patches in `step S2`. Observe this remains true for floating-point representations of coordinates although one tends to forget real numbers in the mathematical sense are an idealization of the actual grossly limited situation. The apparently minor distinction between points *on* and points *near* an arc or patch makes all the difference.

Consider, for example, the computation of the patches contributed by a ball $b$. The triangulation is computed by taking the constrained convex hull and then selecting an appropriate collection of triangles from the convex hull boundary. Owing to numerical inaccuracies, it may not be possible to force all the arc edges to appear on the boundary, although ideally this should not happen. Such a missing arc edge is part of the approximation of a boundary arc. The same edge is also used in the triangulation of the adjacent patch, the one sharing the arc. If the edge is dropped in one triangulation and not the other, it will be impossible to glue the two triangulations along shared arcs.

In a way, the problem posed by two inconsistent data structures mirrors the common difficulties in writing robust geometric software based on floating-point computation. In the case at hand we deal with the inconsistency by putting all our trust on the data structure for $\mathrm{Del}\, B_\alpha$ and forcing the surface triangulations to conform. This would involve perturbing the points on the boundary of a sphere, if necessary, to make sure all arc edges appear on the convex hull boundary. On occasions this implies the necessity to sacrifice the geometric integrity of the surface for the topological correctness. Again the interpretation of a proper surface with slightly imperfect embedding in $\mathbb{R}^3$ seems appropriate. For an exhaustive analysis of such problem cases and possible ways to handle them, see [1].

*Correctness of the triangulation.* We generate a collection of closed surfaces, each separating a component of $\mathrm{Cpx}\, B_\alpha$ from a void or the outside. The total number of surfaces is the number of components plus the number of voids of $\mathrm{Cpx}\, B_\alpha$. To see this assign a surface to the incident component if it is enclosed by the surface, and assign it to the incident void, otherwise. Let $p_i, e_i$ and $t_i$ denote the number of points, edges and triangles of the *ith* surface. The Euler characteristic of a closed surface with genus $g$ is $2-2g$. This implies

$$p_i - e_i + t_i = 2 - 2g_i,$$

where $g_i$ is the genus of the *ith* surface. Since the number of edges is $e_i = 3t_i/2$ we have $t_i = 2p_i - 4 + 4g_i$. The genus of a surface for a component of $\mathrm{Cpx}\, B_\alpha$ that has no voids is the same as the first betti number of the component, $\beta_1$. If the component has voids, then $\beta_1$ is the sum of the genuses of the surfaces for this component. We can compute how many voids and components there are in total and we can sum the individual relations. See [6] for a discussion on betti numbers and on how to compute

the number of voids and components. Summing over all the $k$ components and $l$ voids of $\mathrm{Cpx}\, B_\alpha$, $i = 1, 2, ..., k + l$, we get

$$\begin{aligned} t &= \sum_i t_i \\ &= \sum_i (2p_i - 4 + 4g_i) \\ &= 2p - 4(k + l) + 4\sum_i g_i \end{aligned}$$

The number of components is $k = \beta_0$, the number of voids is $l = \beta_2$ and the sum of genuses is $\sum_i g_i = \beta_1$. Therefore, we have

$$t = 2p - 4(\beta_0 - \beta_1 + \beta_2),$$

where $t$ and $p$ are the total number of triangles and points in the triangulation. The above equation is necessary though not sufficient for the topological consistency of the triangulation. We use it as a simple test for evidence that the collection of points, edges and triangles generated by our algorithm is indeed a triangulation consistent with the dual complex model of the molecule. We also test whether the set of triangles indeed form a 2-manifold (collection of closed surfaces). A necessary and sufficient condition for a 2-manifold is (i) every edge belongs to exactly two triangles, and (ii) the link of every point is a single cycle of edges.

*Parallelization.* Except for `step C2`, all computations can be parallelized in a fairly straightforward manner. Once $\mathrm{Del}\, B_\alpha$ is available, each further step can be executed in parallel by distributing simplices bounding $|\mathrm{Cpx}\, B_\alpha|$ to the available processors in about equal numbers. Triangulations of convex patches are independent for each vertex, triangulations of arcs and saddles are independently for each edge, and triangulations of concave patches are independently for each triangle of $|\mathrm{Cpx}\, B_\alpha|$. Note, however, that arcs are to be approximated before patches can be triangulated, patches are to be shrunk before saddles can be triangulated, and saddles are to be complete before concave patches can be triangulated.

Software described in [2] is used for `steps C2` and `C3`. The first author of this paper implemented `steps S1`, `S2`, `M1`, and `M2` on an SGI challenge system with 12 processors. Table 1 illustrates the experimental performance. The time it takes to compute a surface triangulation from the Delaunay complex is measured in seconds. Constructing the Delaunay complex is considered a preprocessing step and is not included in the statistics. The topological correctness of the triangulation is checked in a post-processing step, and the amount of time for checking is also not included in the statistics. The time complexity of the parallel algorithm, assuming we have already precomputed $\mathrm{Del}\, B_\alpha$ and $|\mathrm{Cpx}\, B_\alpha|$, is $O(k)$ over $n$ processors where $k$ is the average number of simplices in the link of a vertex and $n = |B|$.

Table 1
Timing results for triangulating the surfaces of proteins under the SA and the MS models using 4, 8, and 12 processors of an SGI challenge system. The proteins are gramicidin (G), trypsin (T), myoglobin (M), ribonuclease mutant (R), bactereorhodopsin (B), and hemoglobin thionville (H). The probe radius is 1.2 Å in all cases. In the case of the SA model, the time for `steps S1` and S2 is measured. In the case of the MS model, the time for `steps S1, S2, M1` and `M2` is measured

| | No. of atoms | Model | No. of triangles | Time in seconds 4 p | 8 p | 12 p |
|---|---|---|---|---|---|---|
| | | SA | 12 784 | 2.12 | 1.32 | 1.17 |
| G | 316 | MS | 35 544 | 2.38 | 2.85 | 1.49 |
| | | SA | 25 944 | 4.46 | 2.96 | 2.49 |
| T | 909 | MS | 74 064 | 5.50 | 4.28 | 3.55 |
| | | SA | 43 622 | 7.81 | 5.85 | 5.34 |
| M | 1381 | MS | 119 002 | 9.45 | 7.49 | 6.72 |
| | | SA | 43 898 | 7.88 | 5.08 | 4.37 |
| R | 1815 | MS | 125 570 | 9.62 | 6.40 | 6.05 |
| | | SA | 60 202 | 11.22 | 7.59 | 6.98 |
| B | 3726 | MS | 175 786 | 14.40 | 9.60 | 9.00 |
| | | SA | 201 208 | 49.33 | 31.91 | 25.24 |
| H | 9234 | MS | 586 752 | 55.58 | 36.36 | 31.35 |

## 7. Discussion

This paper describes algorithms for triangulating the surface of a molecule under the space filling (SF), solvent accessible (SA), and the molecular surface (MS) models. The computations are based on the Delaunay complex of the spherical balls representing atoms of the molecule. The generated triangulation is guaranteed to be topologically correct and can be used for further processing, such as the display of cavities and the computation of electro-static charges, see [15]. Other triangulation schemes have the drawback that they cannot correctly handle the reentrant parts of the surface. The triangulation generated by our algorithm can be used to accurately display the molecular surface to the required level of detail.

All algorithms are implemented, and the crucial steps are available on a parallel-processor architecture. The resulting software is used to display all geometric concepts described in this paper in the CAVE virtual reality environment [5].

## References

[1] N. Akkiraju, Molecule surface triangulation from alpha shapes, Ph.D. Thesis, Dept. Comput. Sci., Univ. Illinois, Urbana, Il (1996).

[2] N. Akkiraju, H. Edelsbrunner, M. Facello, P. Fu, E. P. Mücke and C. Varela, Alpha shapes: definition and software, Proceedings of the Workshop on Geometric Software, Geometry Center, Minneapolis (1995).

[3] M. L. Connolly, Analytical molecular surface calculation, J. Appl. Cryst. 16 (1983) 548–558.
[4] M. L. Connolly, Molecular surface triangulation, J. Appl. Cryst. 18 (1985) 499–505.
[5] C. Cruz-Neira, D. Sandin, T. A. DeFanti, R. V. Kenyon and J. C. Hart, The CAVE: audio visual experience automatic virtual environment, Comm. ACM 35 (1992) 64–72.
[6] C. Delfinado and H. Edelsbrunner, An incremental algorithm for Betti numbers of simplicial complexes on the 3-sphere, Comput. Aid. Geom. Design 12 (1995) 771–784.
[7] D. P. Dobkin and M. J. Laszlo, Primitives for the manipulation of three-dimensional subdivisions, Algorithmica 4 (1989) 3–32.
[8] H. Edelsbrunner, The union of balls and its dual shape, Discrete Comput. Geom. 13 (1995) 415–440.
[9] H. Edelsbrunner and E. P. Mücke, Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms, ACM Trans. Graphics 9 (1990) 66–104.
[10] H. Edelsbrunner and E. P. Mücke, Three-dimensional alpha shapes, ACM Trans. Graphics 13 (1994) 43–72.
[11] L. J. Guibas and J. Stolfi, Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams, ACM Trans. Graphics 4 (1985) 74–123.
[12] B. Lee and F. M. Richards, The interpretation of protein structures: estimation of static accessibility, J. Mol. Biol. 55 (1971) 379–400.
[13] F. M. Richards, Areas, volumes, packing, and protein structure, Ann. Rev. Biophys. Bioeng. 6 (1977) 151–176.
[14] A. Varshney, F. P. Brooks and W. V. Wright, Linearly scalable computation of smooth molecular surfaces, IEEE Comput. Graphics Appl. 14 (1995) 19–25.
[15] R. J. Zauhar and R. S. Morgan, Computing the electric potential of biomolecules: application of a new method of molecular surface triangulation, J. Comput. Chem. 11 (1990) 603–622.