Proceedings of the 28th Annual Hawaii International Conference on System Sciences — 1995

Measuring Proteins and Voids in Proteins*

Herbert Edelsbrunner † Michael Facello ‡ Ping Fu § Jie Liang ¶

University of Illinois at Urbana-Champaign and
Hong Kong University of Science and Technology

Abstract

Common geometric models for proteins and other molecules are the space filling diagram, the solvent accessible surface, and the molecular surface. We describe software that computes metric properties of these models, including volume and surface area. It also measures voids or empty space enclosed by the protein, and it keeps track of surface area contributions of individual atoms. The software is based on 3-dimensional alpha complexes and on inclusion-exclusion formulas with terms derived from the simplices in this complex. The software is available via anonymous ftp *at* ftp.ncsa.uiuc.edu.

1 Introduction

The *space filling diagram*, SF, introduced by Lee and Richards [11], models a protein as the union of possibly overlapping spherical balls in $\mathbf{R}^3$, see figure 1. Each ball represents an atom and its size is determined by the van der Waals radius of the atom. A void is a piece of empty space completely surrounded by the balls of the diagram.

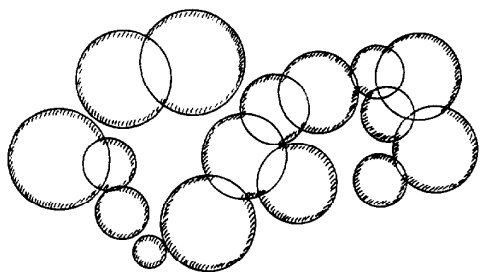

Figure 1: SF: each circular disk represents an atom specified by the location of its center and its van der Waals radius.

The *solvent accessible* model, SA, has been introduced to study the interaction between the protein and *solvent molecules* modeled as spherical balls [11, 13, 14]. The balls representing the solvent molecules are deflated to points and the balls representing atom in the protein are inflated by the same amount, see figure 2. Geometrically, there is little difference between the two models: both are unions of balls, only the sizes differ and thus also the amount of overlap between the balls. A void in the SA model represents all possible locations of centers of captured solvent molecules. It is either contained in a larger void or it lies outside the corresponding SF model.

The *molecular surface* model, MS, is obtained by rolling the sphere representing the solvent molecules over the SF [3, 13, 14]. Alternatively, we can obtain MS from the SA model by removing a layer of solvent radius depth. There is an ambiguity in this definition that occurs when the same piece of space is erased from different directions, see figure 3. We adopt the view of the molecular surface as a possibly self-intersecting 2-dimensional surface in $\mathbf{R}^3$. With this definition, each void in the SA model corresponds to a unique void in the MS model, only that the latter is larger in volume.

*This work is supported by the National Science Foundation, under grant ASC-9200301, the CISE postdoctoral fellowship grant ASC-9404900, and the Alan T. Waterman award grant CCR-9118874. Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the authors and do not necessarily reflect the view of the National Science Foundation.

†Department of Computer Science, University of Illinois at Urbana-Champaign, USA, and Department of Computer Science, Hong Kong University of Science and Technology.

‡Department of Computer Science and National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, USA.

§National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, USA, and Center of Computing Services and Telecommunications, Hong Kong University of Science and Technology.

¶Biophysics Division of the School of Life Sciences, National Center for Supercomputing Applications, and Department of Computer Science, University of Illinois at Urbana-Champaign, USA.

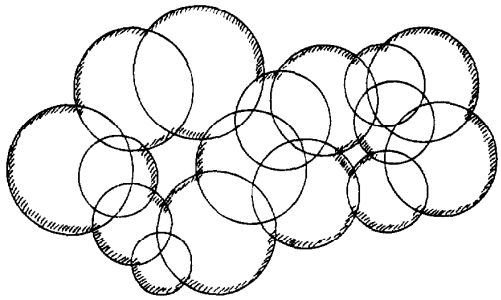

Figure 2: SA: the radius of each disk is the van der Waals radius of the corresponding atom plus the radius of a solvent molecule.

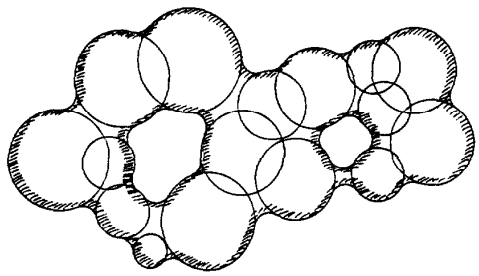

Figure 3: MS: a circle representing solvent molecules rolls about the SF model and bridges narrow cusps and gaps between adjacent atoms.

This paper describes software that computes the metric size of a protein and its voids under the three models. A complete documentation of an earlier version is available in [7]. The software makes no use of the fact that the union of balls is defined by atoms of a molecule. It is therefore more generally applicable to problems about spherical balls in $\mathbf{R}^3$.

The problem of measuring proteins and other molecules has received a fair amount of attention in computational biology and chemistry, and software based on numerical and analytic methods is available, see e.g. [3, 5, 12]. Our software belongs to the category of analytic approaches. Our work differs from earlier work within this category in at least two respects: it is based on the so-called dual complex of a union of balls, see [9], and on inclusion-exclusion formulas with terms for intersections of at most 4 balls at a time, see [6]. The existence of such formulas has been noticed before [10], but no explicit construction was available until recently. In spite of the shallow terms, the formulas are correct (non-approximative) even if there are points covered by many more than 4 balls.

The dual complex is a subcomplex of the larger Delaunay or weighted Delaunay simplicial complex. Our software constructs the latter and represents the dual complex implicitly by marking a subset of the simplices as selected. The simplices that are not selected form a dual representation of the unoccupied space, in the sense used in [1]. Our software uses this representation to compute and measure voids.

In sections 2 and 3 we specify the input to the software, and we describe the various metric properties it computes. The dual complex of a molecule and its relationship to the three models is briefly explained in section 4. The formulas for computing metric properties are expressed in algorithmic language in sections 5, 6, and 7. Section 8 briefly considers the envelope of the molecule, which is the part of space inaccessible from the outside. Section 9 gives a formula for measuring the so-called outside fringe of the molecule. Section 10 discusses how the contributions of individual balls to the surface area can be computed. Section 11 explains the explicit construction of voids in the complex, which is necessary for computing measures of voids in any of the three protein models. Section 12 addresses performance issues of the software. Section 13 concludes this paper with a few remarks placing its contents in the wider context of protein structure modeling and protein dynamics simulation.

2 The data

Each ball is specified by the three coordinates of its center, $x, y, z \in \mathbf{R}$, together with its radius, $w \in \mathbf{R}$. The collection of balls is stored in a linear array, B. The software is based on the notion of the *initial radius* or *weight* of a ball, and a real parameter, α. The parameter globally modifies all radii. For applications where radii do not change, α can be set to zero and henceforth be ignored, as in the case of protein computation where predefined van der Waals radii are used. However, part of the versatility and efficiency of this software stems from the availability of this parameter, and it is instructive to understand how it interacts with the weight. The specification of α changes a ball with initial radius w to one with *actual* radius

$$\sqrt{w^2\mathrm{sign}(w) + \alpha^2\mathrm{sign}(\alpha)}.$$

We refer to a situation where all initial radii are zero as the *unweighted* case.

Negative values for w and for α are possible and admissible. The underlying theory dictates we use squares of w and α to compute the actual radius. The sign function reintroduces the negative sign, if any, that is otherwise lost by squaring. It is even possible and admissible

Proceedings of the 28th Hawaii International Conference on System Sciences (HICSS '95)
1060-3425/95 $10.00 © 1995 IEEE

that $w^2\text{sign}(w)+\alpha^2\text{sign}(\alpha)$ is negative. In this case, the actual radius is imaginary and all measurements ignore this ball as if it were not in B. While this possibility seems to lack the support of a physical explanation, it is essential in a uniform geometric treatment of the entire domain of parameter values.

3 The output

As mentioned in the introduction, both the SF and the SA models of a protein are geometric unions of spherical balls. Our software does not distinguish between these two models and assumes the radii of the balls in B have been assigned appropriately. The MS model is different from SF and SA and needs to be computed separately. We consider four problems: measuring the model (SF, SA, or MS) itself, measuring a single void or the totality of voids, measuring the envelope, and measuring the outside fringe, see tables 1 and 2. For the SF/SA model we compute volume, surface area, arc length, and number of corners. Arcs and corners have no meaning in the MS model, so only volume and surface area are computed.

space filling (SF) or solvent accessible (SA)				
	volume	area	length	corners
ball union	V_s	A_s	L_s	C_s
one void	V_s^v	A_s^v	L_s^v	C_s^v
total voids	V_s^{tv}	A_s^{tv}	L_s^{tv}	C_s^{tv}
envelope	V_s^e	A_s^e	L_s^e	C_s^e
outside fringe	V_s^o	A_s^o	L_s^o	C_s^o

Table 1: The various measurements computed for SF and SA model of a protein.

molecular surface (MS)		
	volume	area
rounded union	V_m	A_m
one void	V_m^v	A_m^v
total voids	V_m^{tv}	A_m^{tv}
envelope	V_m^e	A_m^e
outside fringe	V_m^o	A_m^o

Table 2: The measurements computed for the MS model of a protein.

V_s and A_s denote the volume and the surface area of the union of spherical balls in B, $\bigcup B$. The boundary of $\bigcup B$ consists of spherical patches separated from each other by circular arcs. L_s denotes the total length of all arcs and is thus some kind of 1-dimensional measure of the protein surface. Spherical patches and circular arcs meet at points called corners. C_s denotes the total number of corners.

A *void* is a bounded component of $\mathbf{R}^3 - \bigcup B$. Again we compute the volume, V_s^v, the surface area, A_s^v, the total arc length, L_s^v, and the number of corners, C_s^v. Besides individual voids, we compute the *total* measure of all voids by summing over all voids of $\bigcup B$: $V_s^{tv}, A_s^{tv}, L_s^{tv}, C_s^{tv}$.

The *envelope* of $\bigcup B$ is $\bigcup B$ union its voids. In other words, it is $\bigcup B$ as seen from the outside with all interior void space filled. Its measurements are $V_s^e = V_s + V_s^{tv}$, $A_s^e = A_s - A_s^{tv}$, $L_s^e = L_s - L_s^{tv}$, $C_s^e = C_s - C_s^{tv}$.

Finally, we measure the *outside fringe*. This is the part of $\bigcup B$ that reaches into the unbounded component of the complement of the dual complex. Its measurements, $V_s^o, A_s^o, L_s^o, C_s^o$, are computed using similar formulas as for $\bigcup B$ and the voids. The main reason for measuring the outside fringe is for software verification purposes. In particular, the following relations for the measurements can be used to double-check correctness:

$$\begin{aligned} V_s + V_s^{tv} - V_c^{tv} - V_c - V_s^o &= 0, \\ A_s - A_s^{tv} - A_s^o &= 0, \\ L_s - L_s^{tv} - L_s^o &= 0, \\ C_s - C_s^{tv} - C_s^o &= 0, \end{aligned}$$

where V_c is the volume of the dual complex, and V_c^{tv} is the total volume of the voids in the dual complex. Both are easily computed as sums of tetrahedron volumes.

The measurements taken for the MS model can be classified analogously, except there are no counterparts for arcs and corners. We compute the volume and the surface area for the MS model, its voids, its envelope, and its outside fringe, see table 2. The following relations can be used to check correctness of the results:

$$\begin{aligned} V_m + V_m^{tv} - V_c^{tv} - V_c - V_m^o &= 0, \\ A_m - A_m^{tv} - A_m^o &= 0. \end{aligned}$$

Possible ambiguities in the definitions, in particular of V_m, are clarified in section 6.

4 Geometric background

The software described in this paper assumes the availability of the dual complex $\mathcal{K}$ of $\bigcup B$ as a subcomplex of the weighted Delaunay simplicial complex $\mathcal{D}$ of B. In the discrete and computational geometry literature, $\mathcal{D}$ is sometimes referred to as the regular or the weighted Delaunay triangulation of the ball centers, which are interpreted as points with weights. In the case where all radii are 0, $\mathcal{D}$ is isomorphic to the nerve of the set of Voronoi cells. This section explains these concepts.

Proceedings of the 28th Hawaii International Conference on System Sciences (HICSS '95)
1060-3425/95 $10.00 © 1995 IEEE

The *Voronoi cell* of a point $p \in B$ is the set of points $x \in \mathbf{R}^3$ with Euclidean distance $|xp|$ at least as small as $|xq|$ for any other $q \in B$ [15]. The *nerve* is the system of Voronoi cell collections with non-empty common intersection. Observe that every subset of a collection in the nerve is also in the nerve. For each collection of Voronoi cells in the nerve take the simplex spanned by the points generating the cells in the collection. $\mathcal{D}$ is the set of thus obtained simplices. Vertices in $\mathcal{D}$ correspond to singleton sets in the nerve, edges correspond to pairs, triangles to triplets, and tetrahedra to quadruplets. No collection in the nerve has cardinality beyond 4 if general position is either assumed or simulated [8]. $\mathcal{D}$ is a *simplicial complex*, which technically means that for each $\sigma \in \mathcal{D}$ also the faces of σ belong to $\mathcal{D}$, and every two simplices in $\mathcal{D}$ are either disjoint or meet in a common face. A *subcomplex* of $\mathcal{D}$ is a simplicial complex $\mathcal{K} \subseteq \mathcal{D}$.

For a generalization to non-zero radii, set the *weighted distance* of $x \in \mathbf{R}^3$ to $p \in B$ with radius w equal to $|xp|^2 - w^2$. For each ball we get a (possibly empty) *weighted* Voronoi cell, and $\mathcal{D}$ is again isomorphic to the nerve of the collection of cells.

The *dual complex* $\mathcal{K}$ of $\bigcup B$ is a subcomplex of $\mathcal{D}$. More specifically, it is the subcomplex isomorphic to the nerve of the weighted Voronoi cells restricted to within their respective balls, see figure 4. As a consequence of the definitions, also the boundaries of $\mathcal{K}$ and $\bigcup B$ are closely related. We refer to [6] for a more detailed development of the above concepts and for proofs of some of their properties, including the inclusion-exclusion formulas used in this paper.

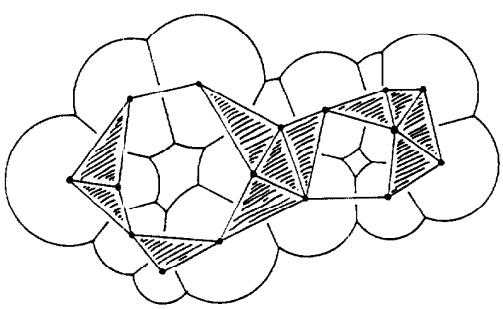

Figure 4: In the plane, the SF and SA models are unions of finitely many disks. The dual complex is a collection of vertices, edges, and triangles.

An interesting aspect of these concepts is related to the parameter α mentioned in section 2. As α increases, all balls grow but the weighted Voronoi cells remain the same. It follows that $\mathcal{D}$ remains unchanged, and $\mathcal{K}$ becomes a larger and larger subcomplex of $\mathcal{D}$, until $\mathcal{K} = \mathcal{D}$ at $\alpha = +\infty$. This in effect defines a sequence of the simplices in $\mathcal{D}$, ordered by value of α at which they enter $\mathcal{K}$. This sequence is referred to as the *filter* of $\mathcal{D}$ and is amenable to the computation of connectivity questions [4]. In particular, it facilitates the efficient construction of the voids in $\mathcal{K}$, as detailed in section 11.

The construction of $\mathcal{D}$, of its filter, and of $\mathcal{K}$ are fairly involved tasks in software systems design. A particularly challenging aspect is the consistent treatment of degenerate cases necessary for robustness. Such a robust system is available via ftp at ftp.ncsa.uiuc.edu and is referred to as the *alpha shape software*. It forms the basis of the software described in this paper, which is now distributed concurrently in one package.

5 Space filling and solvent accessible

Proteins are measured using two types of inclusion-exclusion formulas, the straight and the decomposed ones, both proved in [6]. For the union of balls, $\bigcup B$, we use the straight inclusion-exclusion formulas evaluated in a single loop over the simplices of $\mathcal{K}$.

First, the relevant output parameters, V_s, A_s, L_s, C_s, are initialized to zero. The main loop considers all simplices of $\mathcal{K}$ and computes intersections of 1, 2, 3, and 4 balls, see figure 5. We use volume(.) and area(.) to

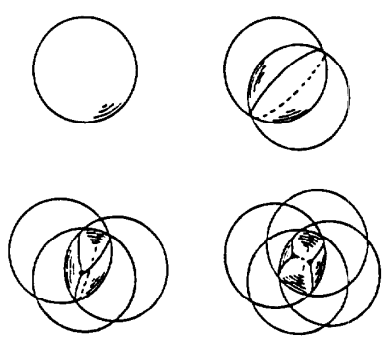

Figure 5: The intersection of 1, 2, 3, and 4 spherical balls in three dimensions.

denote the unambiguously defined volume and surface area of such an intersection. The boundary of an intersection consists of 1, 2, 3, or 4 spherical patches, and length(.) denotes the total length of the 0, 1, 3, or 6 circular arcs separating these patches. The arcs meet at common vertices referred to as *corners*, and for 1, 2, 3, 4 balls there are 0, 0, 2, 4 corners. To simplify the notation, we denote a simplex by the list of its vertices, and we denote a vertex by its index. Without causing any confusion, the same index also denotes the spherical ball centered at this vertex.

Proceedings of the 28th Hawaii International Conference on System Sciences (HICSS '95)
1060-3425/95 $10.00 © 1995 IEEE

```
for each σ ∈ K do
    if σ = i then
        V_s := V_s + volume(i);
        A_s := A_s + area(i)
    endif;
    if σ = ij then
        V_s := V_s − volume(i ∩ j);
        A_s := A_s − area(i ∩ j);
        L_s := L_s + length(i ∩ j)
    endif;
    if σ = ijk then
        V_s := V_s + volume(i ∩ j ∩ k);
        A_s := A_s + area(i ∩ j ∩ k);
        L_s := L_s − length(i ∩ j ∩ k);
        C_s := C_s + 2
    endif;
    if σ = ijkℓ then
        V_s := V_s − volume(i ∩ j ∩ k ∩ ℓ);
        A_s := A_s − area(i ∩ j ∩ k ∩ ℓ);
        L_s := L_s + length(i ∩ j ∩ k ∩ ℓ);
        C_s := C_s − 4
    endif
endfor.
```

It turns out that the evaluation of the straight inclusion-exclusion formulas, as described above, is considerably slower than the computations following the decomposed formulas, see sections 7 and 9. Indeed, it is more efficient to measure $\bigcup B$ using the reduction to voids and outside fringe based on the linear relations in section 3.

6 Molecular surface

Atoms in the MS model have the same size as in the SF model. In spite of the resulting resemblance of the two, the possibly less obvious combinatorial and topological relationship between the MS and the SA models is more useful and exploited in computing the volume and area of the MS model.

Recall that the MS model is obtained by rolling a sphere representing solvent molecules about the SF model. The sphere touches but does not otherwise overlap the SF model. By construction, the center of the sphere lies on the boundary of the SA model. This implies that for each spherical patch of the SA model there is a corresponding smaller spherical patch of the MS model. For each circular arc in the boundary of the SA model there is a torus patch in the MS model. Finally, for each corner of the SA model there is a (concave) spherical patch in the MS model. See figures 2 and 3 for the SA and MS models of the same set of disks in the plane.

The above correspondence suggests we compute the volume of the MS model from the volume of the SA model by subtracting the volume of annulus pieces, solid torus pieces, and ball sectors. The area of the MS model is computed by accumulation of the area of convex spherical patches, torus patches, and concave spherical patches. The convex spherical patches are measured following the same inclusion-exclusion pattern as the area computation in the SA model. The torus patches correspond to circular arcs and their area is computed following the inclusion-exclusion pattern of the length computations in the SA model. Finally, the concave spherical patches are measured following the inclusion-exclusion pattern counting corners of the SA model. Further details are omitted.

As mentioned earlier, there are some ambiguities in the definition of the MS model that need clarification. Since we talked about computing its volume and area, we obviously think of it as a 3-dimensional object. The boundary of this object in general consists of several closed surfaces, and each is possibly self-intersecting but always orientable. All atoms lie on the *inside* of each surface, and the other side, the *outside*, contains no atoms. In contrast to previous conventions, see e.g. [3], we ignore self-intersections and measure the area of the entire surface. The outside is a Riemann space with flaps that overlap in $\mathbf{R}^3$. The volume of the inside is

$$V_{\mathrm{m}} = \int_{x \in \mathbf{R}^3} (1 - \kappa(x)) dx,$$

where $\kappa(x)$ is the number of flaps covering x. In the absence of self-intersections, $\kappa(x) = 0$ inside and $\kappa(x) = 1$ outside. In this case, V_{m} coincides with the conventional notion of volume. In practice, overlapping flaps are rare and necessarily small. It therefore makes little difference whether or not self-intersections are removed. We choose to adopt the Riemannian view as the most elegant of all options. Furthermore, it satisfies the intuitive relations used to double-check correctness, see section 3.

Just as for $\bigcup B$, it is possible to define voids and outside fringe of an MS model. These can be measured by translating the inclusion-exclusion formulas of sections 7 and 9, similar to the above discussion of measuring the MS model itself. The details of this translation are tedious and omitted from this paper.

7 Voids

The union of disks in figure 4 has two voids also shown in figure 6. Each void is contained in a corresponding void of the dual complex. As proved in [6], this holds generally and also in $\mathbf{R}^3$. That is, there is a one-to-one

Proceedings of the 28th Hawaii International Conference on System Sciences (HICSS '95)
1060-3425/95 $10.00 © 1995 IEEE

correspondence between the voids of $\bigcup B$ and the ones of $\mathcal{K}$, and each void in $\bigcup B$ is contained in the corresponding void in $\mathcal{K}$. The volume of the void in $\bigcup B$ is measured by subtracting the pieces of the balls reaching into the void of $\mathcal{K}$. The corresponding theory is expressed by the decomposed inclusion-exclusion formulas. Each term measures the intersection of $1 \leq m \leq 3$ balls and $4-m$ half-spaces. The limit of at most 3 balls per term has a noticeable positive effect on the speed of the evaluation. The planes bounding the half-spaces pass through the centers of the balls so that the influence of the half-spaces can be expressed by multiplying the ball intersection with an angle. It is convenient to measure angles in revolutions, that is, normalized between 0 and 1. The decomposed inclusion-exclusion formulas are used to measure a single void, and by accumulating single void measurements, the total size of the collection of voids.

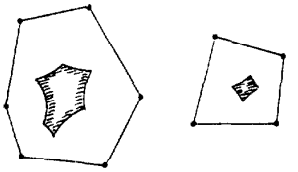

Figure 6: The voids in the disk union of figure 4 are highlighted; each is contained in the corresponding void of the dual complex.

To measure a single void, $\mathcal{V}$, we initialize the volume to the sum of volumes of the tetrahedra of the corresponding void in $\mathcal{K}$; the area, length, and number of corners are initialized to zero. The main loop considers all tetrahedra in $\mathcal{V}$ and their faces, if they are in $\mathcal{K}$. For each such tetrahedron-face pair, it measures a sector, wedge, or a pawn. These are the intersections of balls and half-spaces mentioned above. Following the convention in section 5, a vertex and the ball around this vertex are both denoted by the vertex index. Given four vertices, i, j, k, ℓ, the half-space containing i with bounding plane passing through j, k, and ℓ is denoted by $i_{jk\ell}$. For a tetrahedron $ijk\ell$, $i \cap j_{ik\ell} \cap k_{ij\ell} \cap \ell_{ijk}$ is a *sector*, $i \cap j \cap k_{ij\ell} \cap \ell_{ijk}$ is a *wedge*, and $i \cap j \cap k \cap \ell_{ijk}$ is a *pawn*, see figure 7. Notice that a pawn is exactly half of the intersection of the 3 balls. The area(.) function measures only the spherical patches in the boundary of a sector, wedge, and pawn, and the length(.) function measures only the circular arcs separating spherical patches.

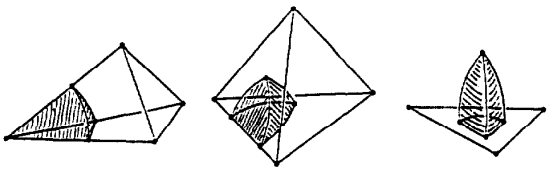

Figure 7: A sector, a wedge, and a pawn.

```
for each σ = ijkℓ ∈ V do
    if i ∈ K then
        V_s^v := V_s^v − volume(i ∩ j_ikℓ ∩ k_ijℓ ∩ ℓ_ijk);
        A_s^v := A_s^v + area(i ∩ j_ikℓ ∩ k_ijℓ ∩ ℓ_ijk)
    endif; do the same for j, k, and ℓ;
    if ij ∈ K then
        V_s^v := V_s^v + volume(i ∩ j ∩ k_ijℓ ∩ ℓ_ijk);
        A_s^v := A_s^v − area(i ∩ j ∩ k_ijℓ ∩ ℓ_ijk);
        L_s^v := L_s^v + length(i ∩ j ∩ k_ijℓ ∩ ℓ_ijk)
    endif; do the same for ik, iℓ, jk, jℓ, and kℓ;
    if ijk ∈ K then
        V_s^v := V_s^v − volume(i ∩ j ∩ k ∩ ℓ_ijk);
        A_s^v := A_s^v + area(i ∩ j ∩ k ∩ ℓ_ijk);
        L_s^v := L_s^v − length(i ∩ j ∩ k ∩ ℓ_ijk);
        C_s^v := C_s^v + 1
    endif; do the same for ijℓ, ikℓ, and jkℓ
endfor.
```

8 Envelopes

As mentioned in section 3, the measurements related to the impact of $\bigcup B$ on its surroundings are sometimes of interest. This is for example the case in the study of nano-crystals, where the volume is defined so it reflects the number of water molecules pushed aside by an invading nano-crystal. These water molecules have no way to access the voids of $\bigcup B$. We therefore measure the union of balls with voids filled. The volume of the envelope is the sum of the volume of $\bigcup B$ and its voids, whereas the area, length, and the number of corners are obtained as differences of these measurements.

9 Outside fringe

The *outside fringe* is the part of $\bigcup B$ that lies outside the dual complex, see figure 8. The situation is similar to the one in section 7, only that we now talk about the *unbounded* component, K_∞, of $\mathbf{R}^3 - \bigcup \mathcal{K}$. K_∞ contains the unbounded component, B_∞, of $\mathbf{R}^3 - \bigcup B$. Formally, $K_\infty - B_\infty$ is the outside fringe.

The following approach to computing its volume, area, length, and number of corners is implemented

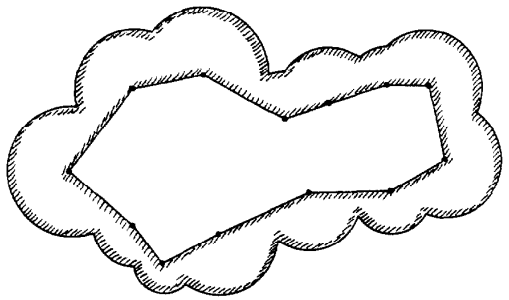

Figure 8: The outside fringe of the disk union in figure 4.

in two different ways. After initializing $V_s^\circ, A_s^\circ, L_s^\circ, C_s^\circ$ all to 0, we accumulate the terms of the decomposed inclusion-exclusion formula. Each vertex, edge, and triangle $\sigma \in \mathcal{K}$ corresponds to possibly several terms collected in one by computing the angle, φ_σ, at σ in K_∞. For σ a vertex, φ_σ is the solid angle in revolutions inside K_∞. For σ an edge, φ_σ is the dihedral angle in revolutions inside K_∞. For σ a triangle, $\varphi_\sigma = 0$ if σ does not bound K_∞, $\varphi_\sigma = \frac{1}{2}$ if σ bounds K_∞ on one side, and $\varphi_\sigma = 1$ if σ bounds K_∞ on both sides.

```
for each σ ∈ K do
    if σ = i then
        V_s° := V_s° + φ_σ * volume(i);
        A_s° := A_s° + φ_σ * area(i)
    endif;
    if σ = ij then
        V_s° := V_s° − φ_σ * volume(i ∩ j);
        A_s° := A_s° − φ_σ * area(i ∩ j);
        L_s° := L_s° + φ_σ * length(i ∩ j)
    endif;
    if σ = ijk then
        V_s° := V_s° + φ_σ * volume(i ∩ j ∩ k);
        A_s° := A_s° + φ_σ * area(i ∩ j ∩ k);
        L_s° := L_s° − φ_σ * length(i ∩ j ∩ k);
        C_s° := C_s° + 2 * φ_σ
    endif
endfor.
```

The two implementations of this approach differ in the way the angles φ_σ are computed. The first method initializes each angle to 1, the full angle, and subtracts angles inside tetrahedra in $\mathcal{K}$ and in voids of $\mathcal{K}$. The second method initializes each angle to 0, the empty angle, and adds angles inside K_∞. One of the reasons to implement both methods is that one tends to be fast when the other is slow. Another is that we can compare the results, and if they match this is evidence that the computations are done correctly.

10 Surface area contributions

The surface of $\bigcup B$ consists of patches of spheres, each being part of the boundary of a ball in B. A single ball may contribute an arbitrary number of such patches, and its *area contribution* is the total area of all its patches. Each term in the inclusion-exclusion formula belongs to a group of 1, 2, 3, or 4 balls, and can be split into the same number of terms, each attributed to a single ball. The total contribution of a single ball is then a partial sum in the inclusion-exclusion formula. When the surface area of $\bigcup B$ is computed, we distribute the terms to the individual balls and keep track of partial sums in the array $\mathtt{A_s}[1..n]$; its ith element accumulates the contributions of ball i. The same thing can be done for voids and for the outside fringe; the corresponding partial sums are accumulated in arrays $\mathtt{A_s^{tv}}[1..n]$ and $\mathtt{A_s^\circ}[1..n]$.

Similar to other measurements, we can cross-check the contributions of individual balls to gain confidence in the computed numbers. The contribution of the same ball or atom should be the same, whether computed using the straight inclusion-exclusion formulas of section 5 or the decomposed formulas of sections 7 and 9. Similarly, the contribution of a ball to $\bigcup B$ should be the same as the sum of the contributions to the voids and the outside fringe. Apart from the equations for individual balls, the following relations are supposed to hold:

$$\begin{aligned} A_s - \sum_{i=1}^{n} \mathtt{A_s}[i] &= 0, \\ A_s^{tv} - \sum_{i=1}^{n} \mathtt{A_s^{tv}}[i] &= 0, \\ A_s^\circ - \sum_{i=1}^{n} \mathtt{A_s^\circ}[i] &= 0. \end{aligned}$$

It is possible, in principle, to compute individual area contributions of balls per void, and also individual volume contributions, or individual length contributions of circles. All these measures seem to be of little interest though, and are thus not implemented at this time.

11 Finding voids

Measuring voids and outside fringe assume their availability as sets of tetrahedra from $\mathcal{D} - \mathcal{K}$. These sets in effect represent the voids and outside of $\mathcal{K}$. This section describes the use of a union-find data structure [2] to construct the voids and outside as a system of disjoint sets.

Proceedings of the 28th Hawaii International Conference on System Sciences (HICSS '95)
1060-3425/95 $10.00 © 1995 IEEE

We take advantage of the linear array, called F for filter, which stores the simplices of $\mathcal{D}$ in sorted order, see section 4. Given an index m, all tetrahedra up to position m in F belong to the corresponding dual complex, $\mathcal{K}$, and all tetrahedra after position m belong to $\mathcal{D} - \mathcal{K}$. Similarly, for triangles, edges, and vertices. Let f be the last index of F and m the index corresponding to $\mathcal{K}$. The voids are computed as follows.

```
for i := f downto m + 1 do
    let σ be the simplex stored in F[i];
    if σ is a tetrahedron then
        add {σ} as a singleton set to the system
    elseif σ is a triangle then
        find the sets that contain the tetrahedra
            sharing σ;
        merge the two sets, if they are different
    endif
endfor.
```

The computation simulates the birth and growth by combination of voids as the parameter α decreases from infinity to zero. One of the sets in the system collects tetrahedra of $\mathcal{D} - \mathcal{K}$ that lie outside $\mathcal{K}$. Whenever the encountered triangle σ belongs to only one tetrahedron, its set is merged with this special set. At completion of the process, each set represents a void except for the special set which represents the outside.

12 Remarks on performance

We break down the task of measuring a protein into several steps and briefly discuss the performance of each.

Assuming the protein is specified in protein data bank (pdb) format, it is a trivial matter to translate it into an input format suitable for the alpha shape software, see section 2. The radius of each ball is either taken directly from a standard table of van der Waals radii or the latter is incremented by the radius of the solvent molecules. A translator is provided as part of our software.

In the first step, the alpha shape software constructs the weighted Delaunay simplicial complex, $\mathcal{D}$, of the data set, B. The number of simplices, $f = \operatorname{card}\mathcal{D}$, depends on the size of B, $n = \operatorname{card} B$, and on the distribution of the balls[1]. For dense distributions common for proteins, f is proportional to n. The time to construct $\mathcal{D}$ depends on the input size, n, and the output size, f, and for $f \leq c \cdot n$, c a constant, it runs in expected time $O(n \log n)$. See [9] for further details and timings for various data sets.

The second step is the generation of the filter and the dual complex, $\mathcal{K}$. The filter is the sequence of simplices in $\mathcal{D}$ ordered by the value of α at which they enter $\mathcal{K}$, see section 4. Computing the value of α takes only constant time per simplex, and sorting takes time $O(f \log f)$. For measuring a protein we are only interested in $\alpha = 0$, that is, we need to separate the simplices with negative value of α (the ones in $\mathcal{K}$) from the ones with positive value of α (the ones in $\mathcal{D} - \mathcal{K}$). In the interest of economical software production, this simplification avoiding sorting has not been implemented.

The third and last step is the evaluation of the inclusion-exclusion formulas, as described in this paper. For each simplex, there is only one term in the straight and at most some constant number of terms in the decomposed formula. Each term involves the intersection of 4 or fewer spherical balls and half-spaces. There are analytic expressions that can be evaluated in constant time to measure this intersection. The entire evaluation thus takes time at most $O(f)$. For practical purposes this is not a very meaningful statement because the constant involved in measuring the intersection of 4 balls is fairly large and has a noticeable effect on the performance. Indeed, we experience a significant improvement in performance when we move to terms of at most 3 balls per term, as in the case of the decomposed inclusion-exclusion formula.

Currently, the software runs on single processor computer architectures only. The evaluation of the terms in the formulas can be done completely independent of each other. This implies that an implementation on a massively parallel architecture can speed up the third step by a factor close to the number of available processors. The same is true for the second step computing α values of simplices. This leaves the construction of the Delaunay simplicial complex in the first step as a possible bottleneck for future efforts to develop a fast parallel version of the alpha shape software.

13 Conclusions

This paper describes a small fraction of the possibilities for using the alpha shape software in modeling and simulating macromolecules, such as proteins. The advantage of using complexes over conventional approaches to modeling proteins directly via surfaces (SF, SA, or

[1] There is a fairly frequent misconception in the applied literature that the construction of 3-dimensional Delaunay simplicial complexes necessarily take time proportional to n^2. This is only true for unlikely distributions of the n balls. In the absolute worst case, f is about $2n^2$, and any algorithm explicitly constructing the f simplices will require time proportional to n^2 or worse. Proteins typically consist of locally dense packings of atoms. Such packings tend to imply Delaunay simplicial complexes with small number of simplices; these can be constructed much faster than worst-case examples.

Proceedings of the 28th Hawaii International Conference on System Sciences (HICSS '95)
1060-3425/95 $10.00 © 1995 IEEE

MS) is the availability of a rich set of quickly accessible information. This includes *proximity information* explicitly present in the structure of the Delaunay simplicial complex, *topological information* in terms of components, tunnels, and voids [4], and *metric information* as described in this paper.

We intend to extend the current software to compute additional information crucial in the study of proteins. This will include pocket structures and electrostatic forces. The targeted applications are the study of ligand-protein docking and the simulation of dynamic behavior of proteins. The latter application requires the development of fast dynamic data structures for Delaunay simplicial complexes and algorithms for derivatives of metric information under infinitesimal motion.

Acknowledgements. We thank Ernst Mücke for his implementation of three-dimensional alpha complexes. Without his dedication to create an always consistent simplicial complex this project could not succeed. We also thank Hai-Ping Cheng, Frederic Richards, and Shankar Subramaniam for their encouragement and for convincing us that computing metric properties of a union of spherical balls does have applications in chemistry, physics, and biology.

References

[1] S. Arizzi, P. H. Mott and U. W. Suter. Space available to small diffusants in polymeric glasses: analysis of unoccupied space and its connectivity. *J. Polymer Science* **30** (1992), 415–426.

[2] T. H. Cormen, Ch. E. Leiserson and R. L. Rivest. *Introduction to Algorithms.* MIT Press, Cambridge, Mass., 1990.

[3] T. H. Connolly. Analytical molecular surface calculation. *J. Appl. Cryst.* **16** (1983), 548–558.

[4] J. A. Delfinado and H. Edelsbrunner. An incremental algorithm for betti numbers of simplicial complexes. *In* "Proc. 9th Ann. Sympos. Comput. Geom., 1993", 232–239.

[5] L. R. Dodd and D. N. Theodorou. Analytic treatment of the volume and surface area of molecules formed by an arbitrary collection of unequal spheres intersected by planes. *Molecular Physics* **72** (1991), 1313–1345.

[6] H. Edelsbrunner. The union of balls and its dual shape. *In* "Proc. 9th Ann. Sympos. Comput. Geom., 1993", 218–231.

[7] H. Edelsbrunner and P. Fu. Measuring space filling diagrams and voids. Rept. UIUC-BI-MB-94-01, Molecular Biophysics Group, Beckman Inst. Univ. Illinois, Urbana, Illinois, 1994.

[8] 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.

[9] H. Edelsbrunner and E. P. Mücke. Three-dimensional alpha shapes. *ACM Trans. Graphics* **13** (1994), 43–72.

[10] K. W. Kratky. The area of intersection of n equal circular disks. *J. Phys. A: Math. Gen.* **11** (1978), 1017–1024.

[11] B. Lee and F. M. Richards. The interpretation of protein structures: estimation of static accessibility. *J. Mol. Biol.* **55** (1971), 379-400.

[12] G. Perrot, B. Cheng, K. D. Gibson, J. Vila, A. Palmer, A. Nayeem, B. Maigret and H. A. Scheraga. MSEED: a program for rapid determination of accessible surface areas and their derivatives. *J. Comput. Chem.* **13** (1992), 1–11.

[13] F. M. Richards. Areas, volumes, packing, and protein structures. *Ann. Rev. Biophys. Bioeng.* **6** (1977), 151–176.

[14] F. M. Richards. Calculation of molecular volumes and areas for structures of known geometries. *Methods in Enzymology* **115** (1985), 440–464.

[15] G. Voronoi. Nouvelles applications des paramètres continus à las théorie des formes quadratiques. *J. Reine Angew. Math.* **133** (1907), 97–178.