EXTRACTION, QUANTIFICATION AND VISUALIZATION OF PROTEIN POCKETS

Xiaoyu Zhang*
Department of Computer Science
California State University San Marcos
San Marcos, CA 92096
**Email: xiaoyu@csusm.edu*

Chandrajit Bajaj
Department of Computer Science
University of Texas at Austin
Austin, TX 78712
Email: bajaj@cs.utexas.edu

Molecular surfaces of proteins and other biomolecules, while modeled as smooth analytic interfaces separating the molecule from solvent, often contain a number of pockets, holes and interconnected tunnels with many openings (mouths), aka molecular features in contact with the solvent. Several of these molecular features are biochemically significant as pockets are often active sites for ligand binding or enzymatic reactions, and tunnels are often solvent ion conductance zones. Since pockets or holes or tunnels share similar surface feature visavis their openings (mouths), we shall sometimes refer to these molecular features collectively as generalized pockets or pockets. In this paper we focus on elucidating all these pocket features of a protein (from its atomistic description), via a simple and practical geometric algorithm. We use a two-step level set marching method to compute a volumetric pocket function $\phi_P(x)$ as the result of an outward and backward propagation. The regions inside pockets can be represented as $\phi_P(x) > 0$ and pocket boundaries are computed as the level set $\phi_P(x) = \epsilon$, where $\epsilon > 0$ is a small number. The pocket function $\phi_P(x)$ can be computed efficiently by fast distance transforms. This volumetric representation allows pockets to be analyzed quantitatively and visualized with various techniques. Such feature analysis and quantitative visualization are also generalizable to many other classes of smooth and analytic free-form surfaces or interface boundaries.

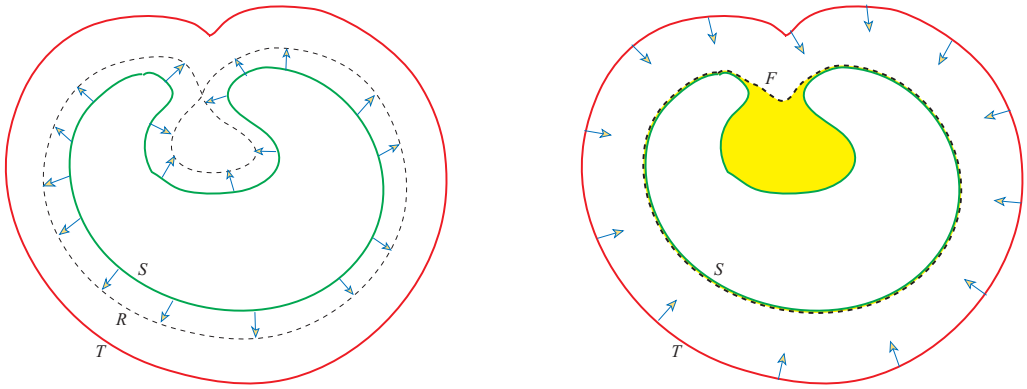
1. INTRODUCTION

Molecular surfaces are solvent contact interfaces between the strongly covalent bonded atoms of the molecule and the ionic solvent environment which is mostly water. Molecular surfaces often contain a number of pockets, holes and interconnected tunnels with many openings (mouths), aka molecular features in contact with the solvent. Several of these molecular features are biochemically significant as pockets are often active sites for ligand binding or enzymatic reactions [7], and tunnels are often solvent ion conductance zones [45]. Since pockets or holes or tunnels share similar surface feature visavis their openings (mouths), we shall sometimes refer to these molecular features collectively as generalized pockets or pockets.

The surface of a protein can be represented as a closed compact surface S in $\mathbb{R}^3$ and the closed interior V as the region bounded by S. It is important to correctly identify the main biophysical features of S in our protein model, such as its "pockets", "tunnels", and "voids" , and so that they can be used for quantitative scoring of binding affinities and other biochemical reactions.

In this paper we present a simple and fast geometric algorithm for extracting pockets of any closed compact smooth surface, particularly complicated solvent contact surfaces of proteins. We use a two-step level-set marching method, first outward from the original protein surface S and then backward from a topological simple enclosing shell obtained as a result of the first marching. The pockets are extracted as the regions outside S and not reached by the backward propagation, as illustrated in Figure 1. The result of the outward and backward propagation is represented as a $3D$ volumetric "*pocket function*" $\phi_P(x)$. The pockets in S can be represented implicitly as the regions $\phi_P(x) > \epsilon$, where ϵ is a small constant. This volumetric representation

*Corresponding author.



(a) (b)

Fig. 1. (a) Outward propagation from S to the shell T that has a simple topology. (b) Backward propagation from T to the final front F. Pockets are extracted as the yellow regions between F and S.

of pockets is very convenient, since it allows us to compute the pocket's mouth surfaces as a contour set $\phi_P(x) = \epsilon$, quantitative the pocket's volumetric properties, and visualize them with various volume visualization techniques.

We present some relevant background and related work in the next section, and then describe details of our pocket extraction algorithm in section 3. In section 4 we discuss our implementation of the algorithm and compare its results with prior published work.

2. BACKGROUND AND RELATED WORK

The description of protein surfaces is important in the analysis of protein-protein and protein-ligand interactions. Several computational models of molecular surfaces for proteins have been used in the past. The van-der-Waals surface [10], is the boundary of the union of balls, where each atom is represented as a spherical solid ball. Other popular models include the solvent accessible surface [38] and the solvent contact surface [14] . More recent work [2, 13, 39, 48]show how to extract triangular meshes of smooth molecular surfaces.

Smooth (and analytic) molecular surfaces can also be modeled as the level set of a volumetric function, e.g. the level set $E(x) = 1$ for the electron density function $E(x)$ in space $\mathbb{R}^3$ [9,18,25]. An isotropic Gaussian kernel function, $G_i(\mathbf{x}) = exp(\frac{b_i*(\mathbf{x}-c_i)^2}{r_i^2} - b_i)$, $\mathbf{x} \in \mathbb{R}^3$, approximates the electron density distribution of a single atom around the atomic center. The decay constant for the Gaussian kernel is denoted by b_i, while c_i is the center of the i-th atom, and r_i is the van der Waals radius of the atom. The electron density field E for a molecule is obtained by summing the individual atomic density distributions for all its atoms,

$$E(\mathbf{x}) = \sum_i G_i(\mathbf{x})$$

Shape properties such as normal, Gaussian and mean curvatures can be computed and displayed for the molecular surfaces [3,18]. However, we are often more interested in the shapes of active sites (pockets) instead of the overall properties of the protein surfaces. Several pocket extraction methods have been developed and published. Delaney [15] uses cellular logic operations on grid points in a spirit similar to our two-step marching algorithm, but its results are very rough approximations and difficult for further visualization and analysis. Edelsbrunner et al. [19] computes pockets for molecular surfaces based on the union-of-balls model using Delaunay triangulations and alpha shapes. The Delaunay triangulation D_B (and its dual Voronoi diagram) are first constructed for the set B of atomic centers [19]. A flow relation can then be defined for two Delaunay tetrahedra, $\tau \in D_B$ and $\sigma \in D_B$, if they share a common plane and the dual Voronoi vertex of τ lies on different sides of the plane from σ. If $\tau \prec \sigma$, τ is called a predecessor of σ and σ a successor of τ. A tetrahedron flows to infinity, if its dual Voronoi vertex is outside D_B or its successor flows to infinity. The alpha-shape $A_B \subset D_B$ at $\alpha = 0$ is the subsimplex of D_B contained in the union of balls. Pockets P are defined

in [19] as the set of Delaunay tetrahedra that do not flow to infinity and do not belong to the alpha-shape A_B, i.e.

$$P \subset D_B - A_B.$$

The alpha-shape based algorithm was implemented and tested for a number of sample proteins [34]. One shortcoming of this method is that the alpha-shape representations of the pockets are usually not smooth. Our algorithm in this paper represents the pockets with a smooth volumetric function, from which smooth pocket surfaces can be computed. Furthermore, our approach works for any representations of protein surfaces, either based on smoothed union-of-balls, [2,3,14,38] or the volumetric model [9, 18]. Our feature analysis algorithm is also clearly generalizable to many other classes of smooth and analytic free-form surfaces.

Complicated shapes are often captured via volumetric functions coupled to morphological operations on the functions. In 2D range images, Krishnapuram and Gupta [30] uses dilation and erosion operations to detect and classify edges; Gil and Kimmel [24] discussed algorithms for computing one-dimensional dilation and erosion operators. In addition to the extraction of polygonal surfaces from volumetric functions, 3D polygonal models are also converted into volumetric representations and then modified, repaired and simplified using morphological operations [21,36].

A related problem to finding pockets in molecular surfaces is shape segmentation, which has been studied using different geometric and topological structures such as shock graphs [40], medial axes [32], skeletons [44], Reeb graphs [26], and others [27, 33, 35]. A notable approach is based on Morse theory, which segments the domain manifold M into stable (unstable) manifolds [16] or Morse-Smale cells [20] of critical points of a Morse function. The Morse function commonly used for shape segmentation is the distance function to a set of discrete points P [16,19,23]:

$$h(\mathbf{x}) = \min_{\mathbf{p} \in P} \|\mathbf{x} - \mathbf{p}\|.$$

Again the Delaunay triangulation (and the dual Voronoi decomposition) can be computed for the points in P. The critical points of h are the intersections of Delaunay elements with their Voronoi complements. The stable-manifolds of the critical points of the distance function to a set of discrete points are called the flow complex in [23], and which is homotopy equivalent to its alpha-shape [17]. The stable manifolds of maxima has the same dimension as the the manifold M and give a segmentation of M. It is possible to consider the pocket extraction problem as the segmentation of the complementary space outside the surface S. However, a large number of points are necessary to sample complex surfaces and a large number of maxima and stable manifolds would segment it into many small pieces that have no direct correspondence to the pockets.

3. ALGORITHMS

In this section, we first present the two-step marching algorithm for computing the pocket function $\phi_P(x)$ in section 3.1. Section 3.2 describes the method of computing signed distance function (SDF) that is based on fast distance transforms and used in the computation of the pocket functions. Section 3.3 discusses the quantitative analysis and visualization of the protein pockets.

3.1. Pocket Extraction

Consider a closed compact surface S, e.g. the green inner curve in Figure 1. We use a two-step marching (fill and removal) strategy to extract pockets in S. First we fill all pockets, voids, and depressions on S by marching outward from S. As shown in Figure 1(a), the front propagates outward from the surface S to a final shell surface T. During the marching the topology of the propagation front changes, for example the topology of front R in Figure 1 is different from S and T. T is chosen to be a propagated front with distance t that is far enough away from S such that T has the simple topology as a sphere and the topology would not change any more by further propagation. The exact value of the distance t from S to T is not significant in our algorithm of pocket extractions. For a typical protein, we choose t as the larger value between 40 $\mathring{A}$ and the twice the largest dimension of the protein.

In the subsequent removal step, the front is propagated backwards from the shell T towards the original surface S. The distance of backward marching is also t so that the front is not allowed to penetrate S and stops when it touches S. Notice the outward

marching in the fill step is irreversible and the final front of the backward marching cannot extend into the depressed regions in the surface S. Therefore in our algorithm, *pockets* are defined as the regions between the final front F of the backward propagation and the original surface S. The shaded (yellow) area in Figure 1(b) illustrates a 2D example pocket found by using this fill and removal strategy. This definition intuitively captures the main characteristics of protein pockets. We now also give a more mathematical definition.

Starting from the initial surface S, the outward propagation front moves along its normal directions at a speed v. The marching front $R(t)$ at time t can be determined according to the level set method [41], i.e. $R(t)$ is the zero level set of a function $\phi(\vec{x}, t)$ satisfying the evolution equation:

$$\phi_t + v|\nabla\phi| = 0$$

with initial condition $\phi(x, t = 0) = d(x)$, where $d(x)$ is the signed distance function (SDF) from S, defined as

$$|d(x)| = \min_{y \in S} |x - y|. \tag{1}$$

SDF $d(x)$ is positive/negative when x is outside/inside the surface S, and the marching front $R(t)$ is the level set $\phi(x, t) = 0$. If the speed $v = 1$ is constant, as we would assume in our two-step marching algorithm, the marching front $R(t)$ at time(distance) t is simply the level set

$$d(x) = t. \tag{2}$$

Assume we already have an efficient algorithm to compute the signed distance functions (SDF) of a closed compact surface, which will be discussed in section 3.2. We present here the algorithm of computing the volumetric *pocket function* $\phi_P(x)$ that represents the pockets in the protein surface.

(1) Compute the signed distance function $d_S(x)$ from the original surface S.
(2) Extract the shell surface T as a level set $d_S(x) = t$, where the distance $t > 0$ is large enough so that T has a simple sphere topology. As mentioned earlier, the exact value of t is not significant in the algorithm.
(3) Compute the signed distance function $d_T(x)$ from the surface T, where the sign of $d_T(x)$ is inverted, i.e. $d_T(x) > 0$ if x is inside T and $d_T(x) < 0$ if x is outside T.
(4) The volumetric pocket function $\phi_P(x)$ is constructed as:

$$\phi_P(x) = \min(d_S(x), d_T(x) - t), \tag{3}$$

where $d_S(x)$ and $d_T(x)$ are the distance functions computed in step 1 and 3. Notice $\phi_P(x) > 0$ only for points outside S and not reachable by backward propagation from T, i.e. points in pockets, tunnels, etc. The bounding surfaces of pockets are then extracted as the level set $\phi_P(x) = \epsilon$, where a small number $\epsilon > 0$ is used to take into consideration the size of solvent atoms. For example, we usually choose ϵ to be between 1 and 1.5 Å.

This pocket extraction algorithm is simple, flexible, and robust. It works for any closed surfaces in $\mathbb{R}^n$ space. Particularly it works for any molecular surface descriptions: union of balls, solvent accessible surface, or contours of electron density functions. Figure 2 shows the successful extraction of two tunnels in an "8" shape.

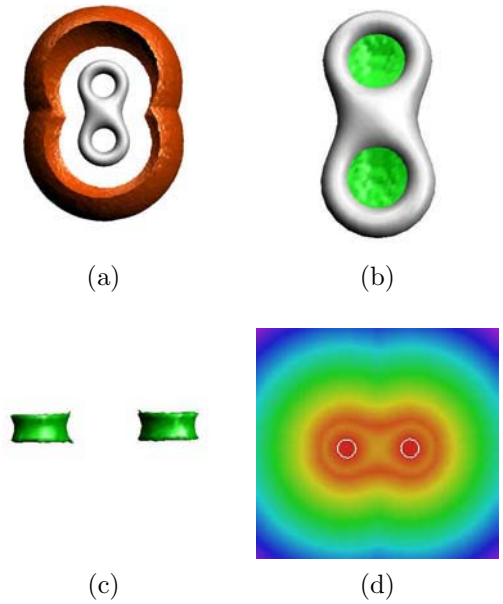


Fig. 2. (a) The original "8" shape in white and the final shell surface from the outward propagation in dark red. (b) The "8" shape and tunnel mouth shown in green. (c) The bounding surfaces of the two tunnels extracted as a level set of the pocket function. (d) A slice of the pocket function of the "8" shape, where the white circles are the cross-section of the tunnel surfaces.

Pocket Mouth In many applications we wish to find a pocket's "mouth", the bounding surface that separates the inside of pockets from the outside region. The number of surface openings (mouths) m of a pocket (or tunnel) classifies the type of the pocket (or tunnel):

- *void* if $m = 0$
- *normal pocket* if $m = 1$
- *hole or simple tunnel (simple conector)* if $m = 2$
- *arbitrary tunnel (multiple connector)* if $m >= 3$

The above pocket function can be used to easily obtain any pocket's mouths. The bounding surfaces of a pocket consists of its mouth and surface patches that are coincident with the original surface S. In other words, pockets mouths are the patches of the final backward propagation front F, which do not match with the original surface S as illustrated in Figure 1(b). Therefore, in our algorithm pocket mouths are determined as portions of the level set $\phi_P(x) = \epsilon$ satisfying the condition $d_S(x) > \epsilon$.

In order to demonstrate the effectiveness of our algorithm, we select a random protein "Bacteriochlorophyll Containing Protein" (PDB ID: 3BCL) from the protein data bank (PDB) [6]. This protein has a very complex molecular surface and contains one large binding site in the middle and some small pockets on its surface, as shown in Figure 3(a). Figure 3(b) shows as a color map a slice of the SDF from the the complex molecular surface of the protein. The cross-section of the protein surface is displayed in Figure 3(b) as white curves, on which the SDF $d(x) = 0$. The large tunnel in the middle is clearly visible, with several small surface pockets and internal voids.

The pocket function of the protein (3BCL) is computed using the algorithm described in section 3.1 and the corresponding slice of the pocket function is shown in Figure 3(c), in which the cross-section of the pocket bounding surface is displayed as white curves. Finally in Figure 3(d), we superimpose the pocket surface with the molecular surface, where pocket mouths are extracted and drawn as yellow line segments. The result matches very well with our own intuition of pockets and their mouths. One can see that our pocket extraction algorithm has almost perfectly located all pockets and holes in the molecular surface.

3.2. Signed Distance Functions

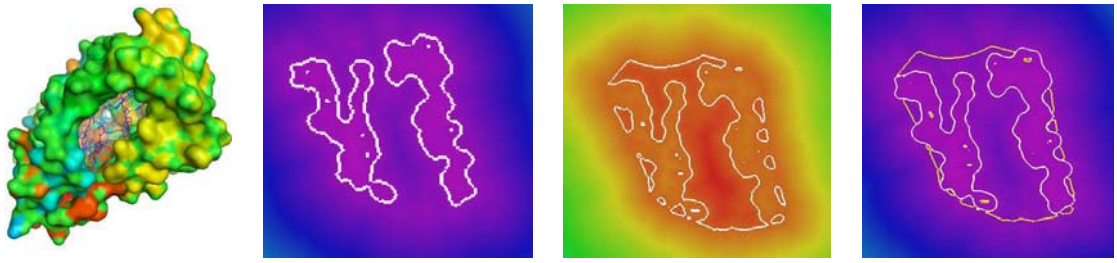
Efficient and stable computation of signed distance functions (SDF) $d(x)$ plays a critical role in the pocket extraction algorithm described in section 3.1. A number of SDF algorithms have been developed in recent years. In this section, we present a method of computing the SDF $d_S(x)$ and $d_T(x)$ based on fast distance transforms [22]. Other stable SDF algorithms may also be applied, for example SDF algorithms using graphics hardware [42] for better speedup.

Given a 2D/3D binary image as input, its distance transform calculates the shortest distance from each pixel (voxel) to the nearest non-zero pixels (voxels). The distance transform computation is very efficient and can be done in time linear to the number of pixels (voxels). We extend the distance transforms to compute SDF for any closed compact surface.

Considering a closed compact surface S embedded in a regular grid, we define a grid point p as a *near point*, highlighted in Figure 4(a), if at least a cell containing p intersects S. Otherwise p is considered as a *far point*. The signed distance function $d_S(x)$ to the surface S is computed as follows:

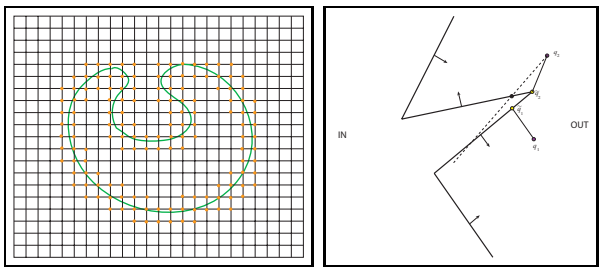
(1) A binary image I_0 is constructed by setting the values of near points to 1 and far points to 0.
(2) We compute the distance transform for the binary image I_0. Particularly, for each far point p its closest near point c_p is recorded. We call c_p the *near cousin* of p. The time for this step is linear to the number of grid points.
(3) For each near point q, its shortest distance $d_S(q)$ to S is computed and the sign of $d_S(q)$ is set positive/negative if q is outside/inside S. The point $\tilde{q}$ on S nearest to the point q is also recorded.

In order to determine whether q is outside or inside S, we assume that S has been decomposed into simplices, e.g. triangles in 3D, and the normal vectors always point towards the outside of S. In $\mathbb{R}^3$, the nearest point $\tilde{q}$ may be inside a triangle, on a triangle edge, or on a triangle vertex. If $\tilde{q}$ belongs to only one simplex $t \in S$, i.e. $\tilde{q}$ is within the interior of t, then q is outside if $(q - \tilde{q}) \cdot \vec{n}_t > 0$, where $\vec{n}_t$ is the normal vector of t. But this dot-product criterion fails if $\tilde{q}$ is a shared point of two or more simplices, i.e. $\tilde{q}$ is on a corner or edge of S. In this case,



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

Fig. 3. Example slices of the pocket function and extracted pocket surfaces for the "Bacteriochlorophyll Containing Protein" (PDB ID: 3BCL). (a) The protein surface and the big tunnel in the middle. (b) A cross-section of the protein surface shown as white curves on the color map of the SDF. (c) A slice of the pocket function for the "Bacteriochlorophyll Containing Protein" is shown as color map and the pocket boundaries are shown as curves. (d) The pockets in (c) are superimposed onto the protein surface in (b).



(a) (b)

Fig. 4. (a) Near points are highlighted in orange. (b) Examples of determining whether a near point is inside/outside the surface S, where $\tilde{q}_1$ is contained in only one simplex but $\tilde{q}_2$ is shared by two simplices. The dashed line is a ray from q_2 to find the nearest intersecting simplex.

we use a ray-shooting method to find the closest simplex to q. We cast a ray R_q from q through an interior point of a simplex containing $\tilde{q}$ and compute the intersection points between R_q and other simplices sharing the same $\tilde{q}$, as illustrated in Figure 4(b). The first simplex t_0 intersected by R_q is chosen and the sign of $d_S(q)$ is set as the sign of $(q - \tilde{q}) \cdot \vec{n}_{t_0}$.

(4) The SDF $d_S(p)$ of a far point p has the same sign as that of its near cousin c_p. The magnitude of $d_S(p)$ is evaluated as $|p - \tilde{c_p}|$, where $\tilde{c_p} \in S$ is the nearst point to c_p on S computed in step 3.

We state two propositions about the signed distance functions $d_S(p)$ computed in the above algorithm.

Prop 3.1. The sign of SDF $d_S(p)$ is correctly set for every far point p .

Proof. We prove this by contradiction. The sign of $d_S(p)$ of the far point p is the same as that of its closest near point c_p. If p is outside S, then its near cousin c_p is inside S. Let us follow the path from p to c_p that consists of three segments along the x, y, and z axes. The last outside point on the path must be a near point and is closer to p than c_p. This contradicts the definition that c_p is the closest near point to p. The same arguments hold if p is inside S. □

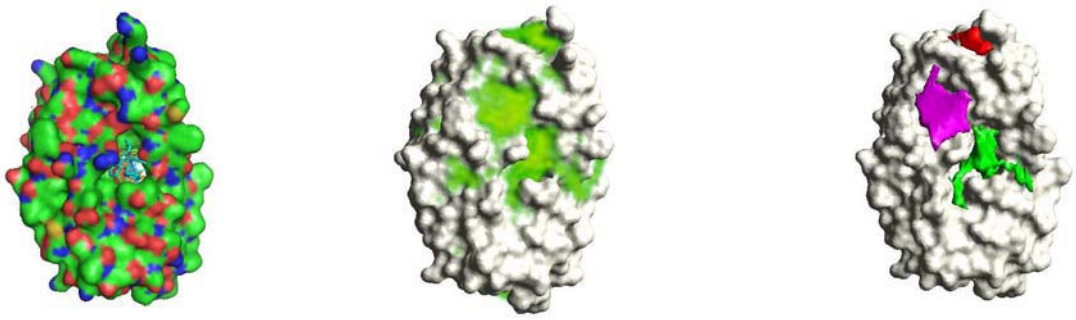
Prop 3.2. The error of $d_S(p)$ is not accumulative and is bounded by the same order as the grid cell side δ.

Proof. Clearly the magnitude of $d_S(p)$ is larger than the distance $|d(p, c_p)|$ from p to its near cousin c_p and less than $|d(p, c_p)| + |d(c_p, \tilde{c_p})|$. The distance $|d(c_p, \tilde{c_p})|$ from the near point c_p to the closest point $\tilde{c_p}$ on S is in the order of $O(\delta)$. Thus we have the following inequality,

$$|d(p, c_p)| \leq |d_S(p)| \leq |d(p, c_p)| + |d(c_p, \tilde{c_p})| = |d(p, c_p)| + O(\delta).$$

Therefore error between $d_S(p)$ and its approximate $d(p, c_p) + d(c_p, \tilde{c_p})$ is bounded by $O(\delta)$. □

Since SDF always achieves the correct signs and has bounded errors for the SDF, the above algorithm is very robust.It is also very efficient and works even for highly complicated protein surfaces. The running time of each step of algorithm is $O(N)$ linear to the number of grid points N, except for step (3). In the worst case, step (3) has computational complexity $O(s \cdot N_n)$, where s is the number of simplices in the surface S and N_n is the number of near points.



(a) (b) (c)

Fig. 5. The molecular surface and pockets of HIV-I protease visualized using combined surface and volume rendering.

However, we use spatial decomposition of the regular grid to limit the search of the nearest point of q to a small subset of simplices on S . On average then, the complexity of step (3) is $O(N_n)$, proportional to the number of simplices in S, which makes the computation of the SDF efficient even for highly complicated protein surfaces in our experiments.

3.3. Quantitative Analysis and Visualization

Representing pockets as a volumetric function $\phi_P(x)$ allows for a number of different ways to visualize and analyze the pocket structures quantitatively.

Visualization As the pocket function $\phi_P(x)$ is a 3D volumetric scalar function, we can visualize it using various volume visualization techniques, e.g. ray-cast or texture based volume rendering and isosurface rendering. As an example of visualization, Figure 5 displays the HIV-I protease (PDB ID: 1HOS), that is important for the maturation of HIV-I virus. An inhibitor can bind in a tunnel of the HIV-I protease, as shown in Figure 5(a). We compute the pocket function of the HIV-1 protease and successfully extract the binding site as the large pocket region of the function. Figure 5(b) renders the pocket function using 3D texture-based volume rendering combined with the protein surface to illustrate the overall distribution of the pocket regions. Figure 5(c) displays the bounding surfaces of the largest four pockets of the HIV-1 protease, one of which is on the other side of the protein and invisible from this view. The ligand binding tunnel is extracted as the pocket (tunnel) with the largest volume. The visualizations were performed using surface and volume rendering capabilities of TexMol [1].

Quantitative Analysis Based on the volumetric pocket function $\phi_P(x)$, we can extract the bounding surfaces of all pockets, tunnels, and voids in a protein once as the level set $\phi_P(x) = \epsilon$. Quantitative measures like the volume and surface area of each pocket can be computed from the pocket function by summing up the contributions from individual cells that belong completely or partially to the pocket. If the 3D domain is decomposed into simplices, the contribution from each simplex to the volume or surface area of the level set $\phi_P(x) = \epsilon$ can be represented as a B-spline function of the variable ϵ and the total measure is the sum of all non-zero B-splines [4].

Additional geometric and shape properties can also be computed for protein pockets based on the pokcet function $\phi_P(x)$, for example curvatures distributions [14, 43], shape histograms [29, 37], coefficients of volumetric function expansions [28], and shape context [5]. Those shape properties of protein pockets may be used for building a database of the proteins pocket structures, and applied to the problem of ligand binding [31]. An affine-invariant method of comparing protein structures is described in [47] by using multi-resolution dual contour trees (MDCT) of the molecular shape functions, e.g. solvent accessibility, combined with geometric, topological, and electrostatic potential properties. We think the pocket functions would better capture the most important features of the protein shapes and provide more accurate comparison and classification.

Contour tree (CT) is an affine-invariant data structure that captures the topological structures of the level sets of a volumetric function $F(x)$ [11], which may also be used for volumetric function matching and protein docking. Each node of the

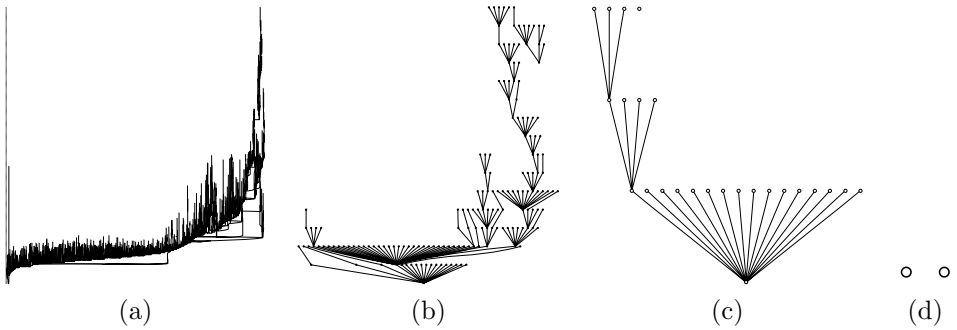


Fig. 6. (a) The contour tree of the pocket function for the "Bacteriochlorophyll A Protein" (3BCL) (b), (c), and (d) are the DCTs of the pocket function at three different resolutions of 16, 4, and 1 intervals.

CT corresponds to a critical point of the function and each arc corresponds to a contour class connecting two critical points. A contour class is a maximal set of continuous contours which do not contain any critical points. If we cut the CT at the isovalue w, the number of connected contours of the level set $F(x) = w$ is equal to the number of intersections (cuts) to CT. In the case of pocket function $\phi_P(x)$, the number of cuts to CT of the pocket function at $\epsilon > 0$ is the number of continuous surface patches bounding the pockets, i.e. the number of separate pockets.

Our pocket algorithm works for general 3D surface models, e.g. the tunnels in the "8" shape shown in Figure2, and for complex protein surfaces. This method is very sensitive. For a complicated surface such as the molecular surfaces computed from electron density functions, the pocket function will capture the large binding sites as well as even small depressions and voids in the surface. This distinctive feature offers both opportunities and challenges in protein shape and structure analysis.

For example, the CT of the pocket function $\phi_P(x)$ for the "Bacteriochlorophyll A Protein (3BCL)" is shown in Figure 6(a). Since we are only interested in the pocket regions where $\phi_P(x) > 0$, the CT has been truncated to remove the uninteresting part of $\phi_P(x) < 0$. However, it still is very complex and contains 2,063 nodes (critical points). A cut at $\epsilon > 0$ would introduce a large number of individual pockets, many of which are very small and of little importance. Furthermore the critical points and the structure of the CT are sensitive to the noise in the data.

Pocket Filtering We need to simplify the pocket function and/or the corresponding CT, in order to focus on the major pockets and filter out small ones. Carr et. al [12] describes a method of simplifying isosurfaces by tagging CT edges with geometric information and suppressing contours of small geometrical measures. While this approach can be applied to the pocket functions to pick the major pockets, the CT itself is not simplified and it is very hard to compare the CT's of two protein pocket functions. Another way of simplification is to construct a simplified data structure for the volumetric function, e.g. the dual contour tree (DCT) introduced in [46, 47].

A DCT studies properties of interval volumes within a specific range of a scalar function and is constructed by partitioning arcs of a CT into sets of connected segments, each of which corresponds to a connected interval volume of the function domain [47]. These interval volumes represent connected regions whose function values are within a specific range. Each node of a DCT is such a connected interval volume. For example, in the case of a pocket function $\phi_P(x)$, the interval volumes within the range $[\epsilon, \max(\phi_P(x))]$ are the 3D regions inside the pockets, where $\epsilon > 0$ is a value for the considerations of the solvent size. The relevant range $[\epsilon, \max(\phi_P(x))]$ of the pocket function is divided into a number of smaller intervals to get a high-resolution and more complete representation of the underlying pocket function. Figure 6(b) shows the DCT constructed from the CT in Figure 6(a) by dividing the $\phi_P(x)$ functional range $[\epsilon, \max(\phi_P(x))]$ into 16 intervals. Each node in Figure 6 (b) represents a connected volume within a certain functional interval. For each node of the DCT, geometric and topolog-

Table 1. Computational time for some examples. T1, T2, and T3 are the time for computing $d_S(x)$, $d_T(x)$, and pockets function respectively.

data	tri#	T1(s)	T2(s)	T3(s)	total (s)
"8" shape	1,536	2.1	5.45	0.33	7.88
"Bacteriochlorophyll A Protein" (3BCL)	275,456	10.25	6.38	0.33	16.96
"Hydrolase" (1C2B)	268,876	9.92	5.63	0.45	16

ical properties of the corresponding interval volume are computed. We refer to [46, 47] for details of constructing DCT and computing the volume and other attributes of the DCT nodes.

Because protein surfaces are highly complicated, they usually contain many small pockets and voids. On the other hand, biologically important active/binding pockets must have enough size to hold the solvated ligand. We can thus remove the very small pockets from further consideration by pruning the DCT nodes whose volumes fall under a given threshold. The DCT in Figure 6(b) has been simplified by pruning. The pruning process can be facilitated by merging functional ranges and constructing DCT's of coarse resolutions. Figure 6(c) and (d) show the DCTs of the same protein with four range intervals and one range interval respectively. A node in a lower-resolution DCT is merged from multiple child nodes from the higher resolution DCT. Pruning a lower resolution DCT node shall remove all its child nodes as well. In the single-range DCT in Figure 6(d), only two nodes are left after prune, one of which contains more than 94% of total pocket volume and corresponds to the large binding site in the middle of the "Bacteriochlorophyll A Protein (3BCL)".

4. IMPLEMENTATION AND EXAMPLES

We have implemented both the pocket extraction and the SDF computation algorithms in C++ and encapsulated them in our freely available TexMol software [1]. Our implementation is portable across multiple compute platforms. The implementation is very robust and efficient, and can compute pockets for complicated molecular surfaces with multiple thousands of atoms in a few seconds. Excluding the time of extracting the original molecular surfaces, Table 1 shows the computation time without optimization on a DELL Laptop with 1.6 GHz processor and 1GB memory for the "8" shape and two proteins, "Bacteriochlorophyll A Protein" (3BCL) and "Hydrolase" (1C2B), downloaded from the PDB. The "Hydrolase" (PDB ID: 1C2B) is a protein complex containing four similar subunits. We choose the dimensions of the regular grid as $128 \times 128 \times 128$ for the balance between accuracy and the requirements for memory. The smaller the grid size, the more accurate the SDF computation. However it also requires more memory and longer computation time because the distance transform time is linear to the number of grid points. Our experiments show that a $128 \times 128 \times 128$ grid is sufficient for extracting protein pockets. For example, when we increase the grid resolution for the protein 3BCL from $128 \times 128 \times 128$ to $196 \times 196 \times 196$, we still get the same set of pockets and the volume of the largest pocket has changed less than 5%.

In Table 1, T1 is the time for computing the SDF $d_S(x)$ for the original surface S, T2 is the time for extracting the shell surface T and computing the SDF $d_T(x)$, and T3 is the time for constructing the pocket function $\phi_P(x)$ and extracting the pockets. 3BCL and 1C2B have longer T1 than the "8" shape because they have more simplices (triangles) in the original surface. All three data sets have similar time for T2 and T3, which is proportional to grid dimension.

We compared our results to the alpha-shape based "CAST" algorithm [8, 34], using the sample protein list given in [34]. CAST uses the union-of-ball model of proteins. It gets atomic radius for each atom from a PDB file, computes the three-dimensional weighted Delaunay triangulation, and then computes the alpha-shape and the volume and areas of the pockets. The pockets are visualized in CAST by displaying the protein residues around the pocket with different color [8], as shown in Figure 7(c). Contrary to the CAST algorithm, our pocket algorithm can use any molecular surface model, including the union-of-ball model. In our implementation, we model molecular surfaces as smooth level sets of the electron density functions $E(x)$, which are computed as the summation of the Gaussian kernel density functions of all atoms contained in the corresponding PDB files. The molecular surface S,

as mentioned before, is extracted as the level set of $E(\mathbf{x}) = 1$. The pocket function $\phi_P(x)$ is computed as described in section 3.1. As mentioned earlier, we can apply more flexible and powerful visualization to the pocket function than the results of CAST. We can actually extract and visualize the pocket volumes themselves, as shown in Figure 7 (a) and (b), which can be further used to be compared with the shapes of binding ligands.

In our analysis, We use multi-resolution DCT's to prune geometrically insignificant nodes and to remove small pocket components. We set a conservative threshold to be 1% of the total pocket volume, since pocket below this threshold are simply too small to be a binding site. A pocket or void is discarded if its volume is below the threshold. For most proteins under consideration, there is a pocket with dominant volume corresponding to the binding site. In table 2, we present the number of pockets after our pruning step, the volume of the largest pocket, and the percentage of its volume over the total pocket volume. In the computation we have chose $\epsilon = 1.5$ for extracting pockets as the level set of $\phi_P(x) = \epsilon$. The results are compared to those from CASTp [8].

Although we used a different molecular surface model from the "CAST" algorithm, our quantitative results are correlated well with the results from CAST [34] and experiments. For example, the Bacteriochlorophyll A Protein (3BCL) was shown in table 2 to contain a large pocket (tunnel) of 94.4% of the total pocket volume, consistent with the experimental binding site and the result in CAST [34]. However, the values of the pocket volumes do not match exactly. This may be due to two major reasons: first the two algorithms use different protein models for pocket extraction; second pockets are segmented differently in the two methods. For example "porin" (2por) has 45 pockets in CAST compared to 2 in our algorithm. We believe our algorithm is quantitatively more accurate according to the visualizations. Our definition of pockets is mathematically rigorous and the extraction algorithm has been shown to visually correct as in Figure 3 and 5.

5. CONCLUSION

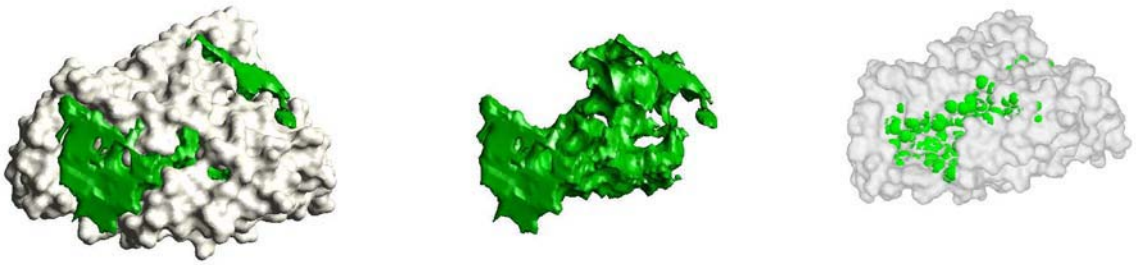
In this paper we present a simple and practical geometric algorithm to compute pockets of any closed compact surface, particularly complicated molecular surfaces. The pockets are represented as a volumetric pocket function, which has the advantage of allowing a wide range of quantitative analysis and visualization. Furthermore, the advantage of our method lies in its generality and applicability to any definition of molecular surfaces. We also present an efficient volumetric sign distance function calculation, necessary for the pocket function. Additionally, we combine quantitative analysis with DCT's to filter insignificant features from molecular surfaces. The combined set of algorithms provide an efficient and robust to extract complementary space features from very complex protein surfaces and additionally other free-form surfaces. The results of our implementation capture all the protein pockets and correlate well with experiments and prior pocket extraction algorithms.

ACKNOWLEDGEMENTS

We thank Dr. Bong-Soo Sohn for several helpful discussions related to pocket computations. This research was supported in part by NSF grants EIA-0325550, CNS-0540033, and NIH grants P20-RR020647, R01-GM074258, R01-GM073087, R01-EB004873. The TexMol software can be freely downloaded from http://www.ices.utexas.edu/CCV/software/

References

1. Bajaj, C., Djeu, P., Siddavanahalli, V., and Thane, A. Texmol: Interactive visual exploration of large flexible multi-component molecular complexes. *Proc. of the Annual IEEE Visualization Conference* (2004), 243–250.
2. Bajaj, C., Lee, H. Y., Merkert, R., and Pascucci, V. Nurbs based b-rep models from macromolecules and their properties. In *In Proceedings Fourth Symposium on Solid Modeling and Applications, Atlanta,Georgia, 1997,C. Hoffmann and W. Bronsvort Eds., ACM Press.* 1997, pp. 217–228.
3. Bajaj, C., and Siddavanahalli, V. An adaptive grid based method for computing molecular surfaces and properties. ICES Technical Report TR-06-57, 2006.
4. Bajaj, C. L., Pascucci, V., and Schikore, D. R. The contour spectrum. In *IEEE Visualization Conference* (1997), pp. 167–173.
5. Belongie, S., Malik, J., and Puzicha, J. Shape matching and object recognition using shape con-



(a) (b) (c)

Fig. 7. Visualizations for the pockets of the "Bacteriochlorophyll A Protein" (3BCL). (a) and (b) show the big central pocket (tunnel) extracted using our algorithm, with and without the protein surface. (c) is the visualization from CASTp [8].

Table 2. Pocket statistics of some sample proteins.

Protein name	PDB ID	# of pockets	largest pocket ($\AA^3$)	percentage	CAST largest pockets ($\AA^3$)
staphylococcal nuclease	1snc	7	538.4	72.7%	757.9
HIC-1 protease	1hvi	8	879	62.8%	1446.8
Endonuclease	2abk	4	2129	83.9%	886.1
acetylcholinesterase	1ack	8	1591	35.4%	1812.3
porin	2por	2	6508.8	95.2%	2306.9
ribonuclease	1rob	3	842.3	78.8%	477.5
thioredoxin reductase	1tde	5	3985	89%	2993.2
NADH peroxidase	2npx	3	6759	92.3%	2846.1
bacteriochlorophyll A protein	3bcl	2	7742	94.4%	11063
glycogen phosphorylase	1gpd	9	7168	63.9%	9059.6
porcine pancreatic elastase	3est	9	1532	62.3%	741.8
elastase with TFA-Lys-Pro-ISO	1ela	10	696	37.6%	304.3
FKBP-FK506	1fkf	5	446	84.3%	292.9

texts. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 4 (2002), 509–522.

6. Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, P. E. The protein data bank. *Nucleic Acids Res 28*, 1 (2000), 235–42. 0305-1048 Journal Article.
7. Berman, J. Structural properties of acetylcholinesterase from eel electric tissue and bovine erythrocyte membranes. *Biochem 12* (1973), 1710.
8. Binkowski, T. A., Naghibzadeh, S., and Liang, J. CASTp: Computed Atlas of Surface Topography of proteins. *Nucl. Acids Res. 31*, 13 (2003), 3352–3355.
9. Blinn, J. F. A generalization of algebraic surface drawing. *ACM Transactions on Graphics* (1982).
10. Branden, C., and Tooze, J. *Introduction to Protein Structure: Second Edition*. Garland, New York, 1999.
11. Carr, H., Snoeyink, J., and Axen, U. Computing contour trees in all dimensions. *Computational Geometry: Theory and Applications 24*, 2 (2003), 75–94.
12. Carr, H., Snoeyink, J., and van de Panne, M. Simplifying flexible isosurfaces using local geometric measures. In *VIS '04: Proceedings of the conference on Visualization '04* (Washington, DC, USA, 2004), IEEE Computer Society, pp. 497–504.
13. Cheng, H.-L., and Shi, X. Quality mesh generation for molecular skin surfaces using restricted union of balls. In *IEEE Visualization 2005* (2005), pp. 399– 405.
14. Connolly, M. L. Analytical molecular surface calculation. *Applied Crystallography 16* (1983), 548–558.
15. Delaney, J. S. Finding and filling protein cavities using cellular logic operations. *J. Mol. Graph. 10*, 3 (1992), 174–177. 159108.
16. Dey, T. K., Giesen, J., and Goswami, S. Shape segmentation and matching with flow discretization. In *Workshop on Algorithms and Data Structures* (2003).
17. Dey, T. K., Giesen, J., and John, M. Alpha-shapes and flow shapes are homotopy equivalent. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing* (2003), pp. 493–502.
18. Duncan, B. S., and Olson, A. J. Shape analysis of molecular surfaces. *Biopolymers 33* (1993), 231–238.
19. Edelsbrunner, H., Facello, M., and Liang, J. On the definition and the construction of pock-

ets in macromolecules. *Discrete Applied Mathematics* (1998), 83–102.

20. Edelsbrunner, H., Harer, J., Natarajan, V., and Pascucci, V. Morse-smale complexes for piecewise linear 3-manifold. In *Proceedings of the nineteenth annual symposium on Computational geometry* (2003), pp. 361 – 370.
21. El-Sana, J., and Varshney, A. Topology simplification for polygonal virtual environments. *IEEE Transactions on Visualization and Computer Graphics 4*, 2 (1998), 133–144.
22. Felzenszwalb, P. F., and Huttenlocher, D. P. Distance transforms for sampled functions. Tech. Rep. Technical Report TR2004-1963,, Cornell University, 2004.
23. Giesen, J., and John, M. The flow complex: a data structure for geometric modeling. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms* (2003), pp. 285–294.
24. Gil, J. Y., and Kimmel, R. Efficient dilation, erosion, opening, and closing algorithms. *IEEE Trans. Pattern Anal. Mach. Intell. 24*, 12 (2002), 1606–1617.
25. Grant, J., and Pickup, B. A gaussian description of molecular shape. *Journal of Physical Chemistry 99* (1995), 3503–3510.
26. Hilaga, M., Shinagawa, Y., Kohmura, T., and Kunii, T. Topology matching for fully automatic similarity estimation of 3d shapes. In *Siggraph 2001* (Los Angeles, USA, 2001), pp. 203–212.
27. Katz, S., and Tal, A. Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM Transactions on Graphics 22*, 3 (2003), 954–961.
28. Kazhdan, M., Funkhouser, T., and Rusinkiewicz, S. Rotation invariant spherical harmonic representation of 3d shape descriptors. In *Proceedings of the Eurographics/ACM SIGGRAPH symposium on Geometry processing* (2003), Eurographics Association, pp. 156–164.
29. Kazhdan, M., Funkhouser, T., and Rusinkiewicz, S. Shape matching and anisotropy. *ACM Trans. Graph. 23*, 3 (2004), 623–629.
30. Krishnapuram, R., and Gupta, S. Morphological methods for detection and classification of edges in range images. *Journal of Mathematical Imaging and Vision 2*, 4 (Dec. 1992), 351–375.
31. Lee, C., and Varshney, A. *Computing and Displaying Intermolecular Negative Volume for Docking.* Springer Berlin Heidelberg, 2006.
32. Leymarie, F. F., and Kimia, B. B. The shock scaffold for representing 3d shape. In *Proceedings of the 4th International Workshop on Visual Form*, Lecture Notes In Computer Science. Springer-Verlag, 2001, pp. 216 – 228.
33. Li, X., Woon, T. W., Tan, T. S., and Huang, Z. Decomposing polygon meshes for interactive applications. In *Proceedings of the 2001 symposium on Interactive 3D graphics* (2001), pp. 35–42.
34. Liang, J., Edelsbrunner, H., and Woodward, C. Anatomy of protein pockets and cavities: measurement of binding site geometry and implications for ligand design. *Protein Sci 7*, 9 (1998), 1884–97. 0961-8368 (Print) Journal Article.
35. Mangan, A. P., and Whitaker, R. T. Partitioning 3d surface meshes using watershed segmentation. *IEEE Transactions on Visualization and Computer Graphics 5*, 4 (1999), 308–321.
36. Nooruddin, F. S., and Turk, G. Simplification and repair of polygonal models using volumetric techniques. *IEEE Transactions on Visualization and Computer Graphics 9*, 2 (2003), 191–205.
37. Osada, R., Funkhouser, T., Chazelle, B., and Dobkin, D. Matching 3d models with shape distributions. In *Proceedings of the International Conference on Shape Modeling & Applications* (2001), IEEE Computer Society, p. 154.
38. Richards, F. M. Areas, volumes, packing and protein structure. *Annu Rev Biophys Bioeng 6* (1977), 151–76. 0084-6589 Journal Article Review.
39. Sanner, M., Olson, A., and Spehner, J.-C. Reduced surface: an efficient way to compute molecular surfaces. *Biopolymers 38*, 3 (March 1996), 305–320.
40. Sebastian, T. B., Klein, P. N., and Kimia, B. B. Recognition of shapes by editing shock graphs. In *International Conference on Computer Vision* (2001), pp. 755–762.
41. Sethian, J. A. A fast marching level set method for monotonically advancing fronts. *PNAS 93*, 4 (1996), 1591–1595.
42. Sigg, C., Peikert, R., and Gross, M. Signed distance transform using graphics hardware. In *IEEE Vis2003* (2003).
43. Sonthi, R., Kunjur, G., and Gadh, R. Shape feature determination usiang the curvature region representation. In *Proceedings of the fourth ACM symposium on Solid modeling and applications* (1997), ACM Press, pp. 285–296.
44. Sundar, H., Silver, D., Gagvani, N., and Dickinson, S. Skeleton based shape matching and retrieval. In *Shape Modelling and Applications Conference* (May 2003).
45. Unwin, N. Refined structure of the nicotinic acetylcholine receptor at 4 a resolution. *J. Mol. Biol. 346* (2005), 967–989.
46. Zhang, X., Bajaj, C., and Baker, N. Affine invariant comparison of molecular shapes with properties. Tech. rep., University of Texas at Austin, 2005.
47. Zhang, X., Bajaj, C. L., Kwon, B., Dolinsky, T. J., Nielsen, J. E., and Baker, N. A. Application of new multi-resolution methods for the comparison of biomolecular electrostatic properties in the absence of global structural similarity. *SIAM Multiscale Modeling and Simulation 5* (2006), 1196–1213.
48. Zhang, Y., Xu, G., and Bajaj, C. Quality meshing of implicit solvation models of biomolecular structures. 510–530.