# Computing Elevation Maxima by Searching the Gauss Sphere* 

Bei Wang ${ }^{1}$, Herbert Edelsbrunner ${ }^{1,2}$, and Dmitriy Morozov ${ }^{1}$<br>${ }^{1}$ Department of Computer Science, Duke University, Durham, North Carolina, USA<br>${ }^{2}$ Geomagic, Research Triangle Park, North Carolina, USA


#### Abstract

The elevation function on a smoothly embedded 2-manifold in $\mathbb{R}^{3}$ reflects the multiscale topography of cavities and protrusions as local maxima. The function has been useful in identifying coarse docking configurations for protein pairs. Transporting the concept from the smooth to the piecewise linear category, this paper describes an algorithm for finding all local maxima. While its worst-case running time is the same as of the algorithm used in prior work, its performance in practice is orders of magnitudes superior. We cast light on this improvement by relating the running time to the total absolute Gaussian curvature of the 2 -manifold.


## 1 Introduction

This paper introduces a new algorithm for computing all local maxima of the elevation function defined on a 2 -manifold embedded in $\mathbb{R}^{3}$. This function has been introduced by Agarwal et al. [4] for the purpose of improving the prediction of protein interaction through docking. The approach identifies protrusions (knobs) and cavities (wells) on the two surfaces and matches them up. This idea goes back to Connolly [12] who used a function that maps each point of the protein surface to the fraction of a fixed-radius sphere centered at the point that lies outside the protein volume. As shown by Cazals et al. [6], this function resembles the mean curvature at the point in the limit, when the radius approaches zero. The fixed radius makes a choice of the scale the function reflects.

The elevation function introduced in [4] serves the same purpose, but in contrast to Connolly's function, the elevation is scale independent and marks small as well as large protrusions of varying shape and direction. Its construction is based on the persistence structure of the 2-parameter family of height functions, as explained in the next section. The task at hand is then the computation of all local maxima for two proteins and the use of the type, size, and location of the marked topographic features to identify promising positions for interaction. The experimental study in [23] shows that this approach is effective in finding initial positions that can then be refined by local optimization. The computationally most expensive step in this study is the determination of the elevation maxima. Using the algorithm in [4], the running time for a triangulated 2-manifold with

[^0]$m$ edges is proportional to $m^{5} \log _{2} m$. Since typical proteins give rise to surfaces with hundreds of thousands of edges, the quintic dependence on $m$ is a serious drawback that limits the practical deployment of the method.

In this paper, we give a new algorithm that is faster for triangulated surfaces approximating smooth surfaces that we typically find in practice. They are characterized by having dihedral angles at edges that are close to half the full angle (molecular skin surface [14]). We relate the running time of our algorithm to the total absolute Gaussian curvature of the surface and this way determine that we can expect roughly a ten-thousand fold improvement over the running time of the old algorithm. We note, however, that we offer no improvement in the worst-case performance.

Since we incorporate the surface complexity in terms of total absolute Gaussian curvature into the analysis of the algorithm, it is worth mentioning that there is a large literature on the notion of curvatures for triangulated surfaces. We refer to [2] and [17|22] for details.

Outline. In Section2, we introduce the geometric and topological background needed to understand the elevation function. We do this in two steps, discussing the mathematically cleaner smooth case in Section 2.1 and the computationally more useful piecewise linear (PL) case in Section 2.2. In Section 3 we present the algorithm for computing all elevation maxima, along with some implementation details and the analysis. In Section 4 we present our experimental results, employing our software to compute elevation maxima for a number of triangulated protein surfaces. We gather statistics on critical regions, pairwise intersections, and elevation maxima. We use these statistics as evidence that our assumption is a reasonable approximation of the reality for our data and that the new algorithm runs about four orders of magnitude faster than the old one.

## 2 Preliminaries

### 2.1 The Smooth Case

Morse functions. The class of smooth, real-valued functions is a challenging object that simplifies considerably if we add genericity as a requirement. Letting $f: \mathbb{M} \rightarrow \mathbb{R}$ be a smooth function on a 2 -manifold, a point $x \in \mathbb{M}$ is critical if the derivative at $x$ equals zero. The value of $f$ at a critical point is a critical value. All other points are regular points and all other values are regular values of $f$. A critical point is nondegenerate if the Hessian, that is, the matrix of second partial derivatives at the point is invertible. In this case, the matrix has two non-zero eigenvalues, $\lambda_{1} \neq \lambda_{2}$, and the index of the non-degenerate critical point is the number of negative eigenvalues. A nondegenerate critical point of index 0 is a minimum, of index 1 is a saddle, and of index 2 is a maximum. Finally, $f$ is a Morse function if all its critical points are non-degenerate and its values at the critical points are distinct. Given a value $a \in \mathbb{R}$, the corresponding sublevel set consists of all points with value at most $a, \mathbb{M}_{a}=f^{-1}(-\infty, a]$. Sweeping the manifold in the direction of increasing function value, we get a 1-parameter family of sublevel sets. The topology of the sublevel set changes precisely when the sweep passes through a critical point. Let $t_{1}<t_{2}<\ldots<t_{n}$ be the ordered sequence of critical values and $-\infty=s_{0}<s_{1}<\ldots<s_{n}=\infty$ a sequence of interleaved values,
that is, $s_{i}<t_{i+1}<s_{i+1}$, for all $i$. By assumption of $f$ being Morse, we get from the sublevel set at $s_{i}$ to the one at $s_{i+1}$ by passing exactly one non-degenerate critical point. The change can be characterized in terms of the dimension of the handle we attach to go from $\mathbb{M}_{s_{i}}$ to $\mathbb{M}_{s_{i+1}}$. For index 0 , we add a 0 -handle, that is, an isolated point which we then thicken to a disk. For index 1, we add a 1 -handle, that is an interval attached to the boundary of the sublevel set at its endpoints which we then thicken to a strip. Finally, for index 2, we add a 2 -handle, that is, a disk attached to the boundary of the sublevel set along its boundary circle.

Persistent homology. Looking at the homology groups [18] of the sequence of sublevel sets, we use the concept of persistence to measure the lengths of the intervals along which homology classes exist [15]. Since sublevel sets between two contiguous critical values are indistinguishable, we may consider the finite sequence

$$
\emptyset=\mathbb{M}_{0} \subseteq \mathbb{M}_{1} \subseteq \ldots \subseteq \mathbb{M}_{n}=\mathbb{M}
$$

where we simplify notation by setting $\mathbb{M}_{i}=\mathbb{M}_{s_{i}}$. Fixing a dimension $p(p \geq 0)$, each sublevel set has a $p$-th homology group and the sequence is connected from left to right by homomorphisms induced by inclusion, which we denote as $f_{p}^{i, j}: \mathrm{H}_{p}\left(\mathbb{M}_{i}\right) \rightarrow$ $\mathrm{H}_{p}\left(\mathbb{M}_{j}\right)$. We have a birth at $\mathbb{M}_{i}$ if the map $f_{p}^{i-1, i}$ is not surjective, and we have a death at $\mathbb{M}_{j}$ if the map $f_{p}^{j-1, j}$ is not injective. Furthermore, the death at $\mathbb{M}_{j}$ corresponds to the birth at $\mathbb{M}_{i}$ if there is homology class $\gamma$ in $H_{p}\left(\mathbb{M}_{i}\right)$ that is not in the image of $f_{p}^{i-1, i}$, its image in $\mathrm{H}_{p}\left(\mathbb{M}_{j-1}\right)$ is still not in the image of $f_{p}^{i-1, j-1}$, but its image in $\mathrm{H}_{p}\left(\mathbb{M}_{j}\right)$ is in the image of $f_{p}^{i-1, j}$. We call $f\left(t_{j}\right)-f\left(t_{i}\right)$ the persistence of this birth-death pair. As explained in [8], this method gives a pairing between births and deaths that has many interesting properties. Each death corresponds to a unique birth but not every birth corresponds to a death. To remedy this shortcoming, we extend the sequence of homology groups for extended persistence as described in [9]. Writing $\mathbb{M}^{a}=f^{-1}[a, \infty)$ for the superlevel set of $a$, we go up with absolute homology groups of sublevel sets, as before, and we come back down with relative homology groups,

$$
\begin{aligned}
0 & =\mathrm{H}_{p}\left(\mathbb{M}_{0}\right) \rightarrow \mathrm{H}_{p}\left(\mathbb{M}_{1}\right) \rightarrow \ldots \rightarrow \mathrm{H}_{p}\left(\mathbb{M}_{n}\right) \\
& \rightarrow \mathrm{H}_{p}\left(\mathbb{M}, \mathbb{M}^{n}\right) \rightarrow \ldots \rightarrow \mathrm{H}_{p}\left(\mathbb{M}, \mathbb{M}^{0}\right)=0
\end{aligned}
$$

where we simplify notation by setting $\mathbb{M}^{i}=\mathbb{M}^{s_{i}}, \mathbb{M}^{0}=\mathbb{M}$ and $\mathbb{M}^{n}=\emptyset$. Now every birth corresponds to a death. In fact, we have two events at every critical point, one going up and one coming down, but duality implies that we just get each pair twice, see [9]. As a consequence of duality, the birth-death pairs we get for the negative function, $-f$, are the same. This turns out to be important in the definition of the elevation function.

For 2-manifolds, there is a more elementary way to introduce extended persistence using the Reeb graph of the function. Instead of giving details, we refer to [4] and we mention that this approach leads to a fast algorithm. It consists of constructing the Reeb graph in a sweep [10] followed by deconstructing it in another sweep using cutting and linking trees [4|16]. We run this algorithm for a piecewise linear function on a triangulated 2 -manifold. Letting $m$ be the number of edges in the triangulation, as before, the running time computing the extended persistence for a given height function is bounded by some constant times $m \log _{2} m$.

Elevation. To define elevation, we assume the 2-manifold $\mathbb{M}$ is smoothly embedded in $\mathbb{R}^{3}$. For a direction $u \in \mathbb{S}^{2}$, we consider the height function $h_{u}: \mathbb{M} \rightarrow \mathbb{R}$ defined by $h_{u}(x)=\langle x, u\rangle$. Generically, $h_{u}$ is a Morse function, but for some directions $u$ it is not, either because a critical point is degenerate or because two or more critical points map to the same height value. Considering the entire sphere of directions, we get a 2-parameter family of height functions.

For each $u \in \mathbb{S}^{2}$, we pair up births with deaths using the extended sequence of homology groups defined by the sublevel and the superlevel sets of $h_{u}$. In the Morse function case, each birth-death pair identifies two critical points, $x$ and $y$, one giving birth and the other giving death, and we define the elevation at these two points as their persistence or, equivalently, the absolute height difference in the direction $u, E(x)=$ $E(y)=\left|h_{u}(x)-h_{u}(y)\right|$. Each point of $\mathbb{M}$ is critical in two directions, $u$ and $-u$, and is thus assigned two values, the absolute height difference to the paired critical point in the two directions. Since $h_{-u}=-h_{u}$, the paired point is the same so we get a unique value at every point. This is the elevation function of the 2-manifold, $E: \mathbb{M} \rightarrow \mathbb{R}$.

To get a feeling for this function, we consider a protrusion (a mountain) of the 2 -manifold. To measure the height of the mountain, we measure from the top down, to the first saddle that separates it from an even higher mountain. We can do this in various directions, so we do it to maximize the height. This might be in a direction along which the first saddle is ambiguous. Perhaps there are three such saddles at the same height value in this direction, similar to the third type in Figure 1 in which we have a saddle with the same height difference to three minima. In this direction, we have two violations of genericity required for Morse functions, because there are three critical points with the same height value. Indeed, local maxima of $E$ tend to arise along nongeneric directions. An exception is the 1-legged maximum defined by only two critical points (with one leg between them). Besides this case, we have 2-legged maxima defined by three critical points, and 3-and 4-legged maxima defined by four critical points each; see Figure 1.

Curvature. We will later discover that the running time of our algorithm for finding all local maxima relates to the total absolute curvature of the surface. We introduce this concept using the Gauss map, $N: \mathbb{M} \rightarrow \mathbb{S}^{2}$, defined by mapping a point $x$ of $\mathbb{M}$ to the outer unit normal, $N(x)$, at $x$. Assuming $\mathbb{M}$ is smoothly embedded in $\mathbb{R}^{3}$, the Gauss map is continuous and surjective but not necessarily injective. Indeed, the preimage of $u \in \mathbb{S}^{2}$ consists of all critical points of $h_{u}$ with outer normal $u$, as opposed to $-u$. The multiplicity of $N$ at $u$ and $-u$ together is thus the number of critical points of $h_{u}$. We will see shortly that the total coverage of $\mathbb{S}^{2}$ is exactly the total absolute Gaussian curvature of $\mathbb{M}$.

Letting $x$ be a point of $\mathbb{M}$ and $r>0$ a radius, we define the absolute Gaussian curvature at $x$ by taking the limit of a fraction of areas, $g(x)=\lim _{r \rightarrow 0} \frac{\operatorname{Area}\left(N\left(A_{r}\right)\right)}{\operatorname{Area}\left(A_{r}\right)}$, where $A_{r}$ is the neighborhood of points at distance at most $r$ from $x$ on $\mathbb{M}$. The total absolute Gaussian curvature is the integral of the local quantity, $G(\mathbb{M})=\int_{x \in \mathbb{M}} g(x) \mathrm{d} x$. It should be clear that $G(\mathbb{M})$ is the area of the total coverage of $\mathbb{S}^{2}$, taking multiplicity into account. For a given direction, the multiplicity is $\left|N^{-1}(u)\right|$. Hence, $G(\mathbb{M})=$ $\int_{u \in \mathbb{S}^{2}}\left|N^{-1}(u)\right| \mathrm{d} u$. Writing $c_{\text {avg }}$ for the average number of critical points of the height functions, we thus have the total absolute Gaussian curvature equal to one half times


Fig. 1. The four generic types of local maxima of the elevation function. From left to right: the $1-, 2$-, 3 - and 4 -legged maximum.
the area of the sphere times that average, $G(\mathbb{M})=2 \pi c_{\text {avg }}$. This integral geometry formula for the curvature will come handy in the analysis of our algorithm. For more information on the integral geometry formulation of curvature see Santaló [21].

### 2.2 The PL Case

Triangulated surfaces. We do all computations on a piecewise linear approximation of the smooth 2-manifold. To transport the smooth concepts to the PL category, we think of the PL surface as being approximated by a smooth surface. Tightening the approximation, we get a series and take the limit. This is the general intuition we have in the background guiding the formulation of definitions in the PL case.

A triangulation of a 2 -manifold $\mathbb{M}$ is a simplicial complex, $K$, whose underlying space is homeomorphic, $|K| \approx \mathbb{M}$. It consists of vertices, edges, and triangles. To put $K$ into $\mathbb{R}^{3}$, it suffices to map each vertex to a point; the edges and triangles are the convex hulls (of the images) of their vertices. This is a geometric realization if the triangles meet in shared edges and vertices but not in any other point sets. We call the result a triangulated surface, implicitly assuming that it is geometrically realized in $\mathbb{R}^{3}$. The star of a vertex is the set of simplices that contain it, and the link consists of all faces of simplices in the star that do not belong to the star, $\operatorname{St} v_{i}=\left\{\sigma \in K \mid v_{i} \in \sigma\right\}$; $\operatorname{Lk} v_{i}=\left\{\tau \subseteq \sigma \in \operatorname{St} v_{i} \mid \tau \notin \operatorname{St} v_{i}\right\}$. A PL function $f:|K| \rightarrow \mathbb{R}$ is determined by its values at the vertices. Assuming $f\left(v_{i}\right) \neq f\left(v_{j}\right)$ whenever $i \neq j$, we define the lower link as the subset of simplices in the link where $f$ is smaller than at the vertex, $\mathrm{Lk}-v_{i}=\left\{\sigma \in \operatorname{Lk} v_{i} \mid x \in \sigma \Rightarrow f(x)<f\left(v_{i}\right)\right\}$. Finally, $v_{i}$ is regular if its lower link is contractible, and critical, otherwise. Since $K$ triangulates a 2-manifold, every link is a circle and the only contractible closed subsets are points and closed paths. The lower link of a regular vertex is thus a single vertex or a path connecting two vertices. A minimum is characterized by $\mathrm{Lk}_{-} v_{i}=\emptyset$ and a maximum by $\mathrm{Lk}_{-} v_{i}=\mathrm{Lk} v_{i}$. In the remaining case, the lower link consists of $k+1 \geq 2$ paths and we call $v_{i}$ a $k$-fold saddle, or a simple saddle if $k=1$.

In contrast to the smooth case, it is not possible to turn a $k$-fold into a simple saddle by a small perturbation. We therefore treat them directly, without reduction to simple cases. As an example, consider the Euler-Poincaré Theorem which relates the topology of the 2 -manifold with the critical point structure of its functions. Define the index of a simple critical point as index $\left(v_{i}\right)$, index $\left(v_{i}\right)=0$ if $v_{i}$ is a minimum, 1 if $v_{i}$ is a simple saddle, 2 if $v_{i}$ is a maximum. Assuming $K$ is connected, it is characterized by its genus
and we have $2-2 \cdot$ genus $=n-m+l=\sum_{i}(-1)^{\text {index }\left(v_{i}\right)}$, where $n, m, l$ are the number of vertices, edges, triangles in $K$ and a $k$-fold saddle is represented by $k$ simple saddles in the sum.

Critical regions. Another significant complication we encounter in the PL case is that a vertex is generally critical for an entire region of directions. Letting $h_{u}:|K| \rightarrow \mathbb{R}$ be the height function defined by $h_{u}(x)=\langle x, u\rangle$, the critical region of a vertex is the closure of the set of directions along which $v_{i}$ is critical,

$$
R_{i}=\operatorname{cl}\left\{u \in \mathbb{S}^{2} \mid v_{i} \text { is critical point of } h_{u}\right\} .
$$

We construct it from the closed polygonal curve defined by the star of $v_{i}$. Specifically, we map each triangle in the star to its outer normal direction, a point on $\mathbb{S}^{2}$, and we connect the directions of two neighboring triangles by the shorter of the two connecting greatcircle arcs. This gives a closed polygonal curve, $\pi_{i}$, which may or may not have selfintersections. To cope with the former, more complicated case, we orient $\pi_{i}$ and define the winding number of a direction $u \in \mathbb{S}^{2}$ not on the curve as the number of times the curve goes around the directed line defined by $u$. Viewed along $u$, we count a counterclockwise turn as +1 and a clockwise turn as -1 . Taking the sum we get the winding number, which are denoted as $w\left(u, \pi_{i}\right)$. For detailed study on polyhedron Gauss map, refer to [5]. Examples are shown in Figure 2. The winding number of $u$ relates to the type of the vertex in the height function defined by $u$. Specifically, if $v_{i}$ is regular then the winding number of $u$ is 0 , if $v_{i}$ is a simple critical point then the winding number is $(-1)^{\text {index }\left(v_{i}\right)}$, and if $v_{i}$ is a $k$-fold saddle then the winding number is $-k$.s

Curvature. Thinking of a vertex as a tiny region in an approximating smooth surface, we define its Gaussian curvature as the area of its critical region weighted by the winding number. More useful in this paper is its absolute Gaussian curvature defined as the area weighted by the absolute winding number, $g\left(v_{i}\right)=\int_{u \in \mathbb{S}^{2}}\left|w\left(u, \pi_{i}\right)\right| \mathrm{d} u$. The total absolute Gaussian curvature is then the sum over all vertices, $G(K)=\sum_{i} g\left(v_{i}\right)$. Equivalently, it is the area of the sphere times half the average number of critical vertices, taking multiplicities into account, as usual. The average is taken over all height functions, and we count half the critical vertices because $v_{i}$ is critical for $u \in \mathbb{S}^{2}$ as well as $-u \in \mathbb{S}^{2}$.

## 3 Computation

In this section, we describe how we compute the elevation maxima for a given triangulated surface in $\mathbb{R}^{3}$. The algorithm is straightforward and the only new insight is in the analysis, relating the running time with the total absolute Gaussian curvature of the surface.

Types and filters. Recall that there are four types of elevation maxima for a generic smooth surface, as illustrated in Figure 1. We have the same four cases for a generic triangulated surface $K$ in $\mathbb{R}^{3}$. Each maximum is given by a set of two, three, or four points. We consider the case in which all these points are vertices of $K$. The cases in


Fig. 2. Left: for a direction $u$ with winding number +1 the corresponding vertex appears either as a maximum or a minimum. Right: for winding number -1 we have a simple saddle and for -2 we have a 2 -fold or monkey saddle for the height function defined by the corresponding direction.
which some of the points in $V$ lies on edges of $K$ are similar. Let $V$ be a set of vertices. A necessary requirement for $V$ to define a maximum is that its vertices are critical for a common direction. More specifically, we need them critical in a particular direction that is determined by $V$. This direction, $u_{V}=(y-x) /\|y-x\|$, is slightly different for each type.

1-legged case, $V=\{x, y\}$. Here, $u_{V}$ is the direction defined by the two points.
2-legged case, $V=\left\{x, y_{1}, y_{2}\right\}$. Letting $y$ be the orthogonal projection of $x$ onto the line passing through $y_{1}$ and $y_{2}, u_{V}$ is defined if $y$ lies between $y_{1}$ and $y_{2}$.
3 -legged case, $V=\left\{x, y_{1}, y_{2}, y_{3}\right\}$. Letting $y$ be the orthogonal projection of $x$ onto the plane passing through $y_{1}, y_{2}, y_{3}, u_{V}$ is defined if $y$ lies in the triangle they span.
4 -legged case, $V=\left\{x_{1}, x_{2}, y_{1}, y_{2}\right\}$. Letting $x$ and $y$ be the feet of the shortest line segment connecting the line passing through $x_{1}$ and $x_{2}$ with the line passing through $y_{1}$ and $y_{2}, u_{V}$ is defined if $x$ lies between $x_{1}$ and $x_{2}$ and $y$ lies between $y_{1}$ and $y_{2}$.

PROJECTION FILTER. The direction $u_{V}$ defined by the points in $V$ is defined and belongs to the common intersection of critical regions, $u_{V} \in \bigcap_{v_{i} \in V} R_{i}$.

Note that the non-empty intersection of the critical regions is a necessary but not a sufficient condition for the set $V$ to pass the Projection Filter. In turn, passing the Projection Filter is a necessary but not sufficient condition for the direction $u_{V}$ to be an elevation maximum. For that, the set needs to satisfy another condition. To describe it, we write $x_{0}$ for $x$.

PERSISTENCE FILTER. For each pair $x_{i}$ and $y_{j}$ in $V$, there is an arbitrarily small perturbation $u$ of $u_{V}$ such that $x_{i}, y_{j}$ is a birth-death pair for the height function $h_{u}$.

Algorithm. We compute the elevation maxima in three steps, starting with $2-, 3-, 4$ tuplets $V$ whose points have pairwise overlapping critical regions. The next two steps narrow down the selection using first the Projection and the Persistence Filter.

STEP 0. Compute the critical regions of the vertices of $K$. Letting the critical regions be the nodes of the intersection graph, $R$, we draw an arc if the two regions have a non-empty common intersection. For $k=2,3,4$, let $Q_{k}$ be the set of $k$-cliques, that is, the $k$-tuplets of nodes connected by all $\binom{k}{2}$ arcs. Let $S_{0}=\bigcup_{k} Q_{k}$.
Step 1. Subject each pair, triplet, and quadruplet in $S_{0}$ to the Projection Filter and let $S_{1} \subseteq S_{0}$ be the collection that passes the filter.
STEP 2. Subject each pair, triplet, and quadruplet in $S_{1}$ to the Persistence Filter and let $S_{2} \subseteq S_{1}$ be the collection that passes the filter.

Step 1 and 2 are the same as in [4], so we focus on the implementation of Step 0 in which we compute the 2 -, 3 -, 4 -tuplets with pairwise intersecting critical regions.

Implementation. We break down Step 0 into three smaller steps, constructing the critical regions, finding the intersecting pairs, and computing the cliques of size 2,3 , 4 in the intersection graph. Implementation is done with Perl, C and CGAL [1]. All computations are exact except estimating the area and the bounding box of a critical region.

STEP 0.1. Recall that each critical region, $R_{i}$, is given by a closed polygon with $m_{i}$ edges on the sphere. Those edges may intersect, and we take time $\mathrm{O}\left(m_{i}^{2}\right)$ to construct the decomposition of the sphere [13], including winding numbers for all subregions. Reflecting $R_{i}$ centrally through the origin in $\mathbb{R}^{3}$, we get the region $-R_{i}$ of inward normals along which $v_{i}$ is critical. Constructing all critical regions takes time proportional to $\sum_{i} m_{i}^{2}$.
STEP 0.2. Most critical regions are small and simple. This suggests we use a bounding volume approach to find the intersecting pairs. Specifically, we find an axis-parallel box $B_{i}$ in $\mathbb{R}^{3}$ that encloses the region $R_{i}$ on $\mathbb{S}^{2} \subseteq \mathbb{R}^{3}$. We do this in two steps, first computing the smallest enclosing sphere of $R_{i}$ and second the smallest axis-aligned box that contains the sphere. Assuming that $R_{i}$ fits inside a hemisphere of $\mathbb{S}^{2}$, the smallest enclosing sphere of its vertices also encloses $R_{i}$. To compensate for roundoff errors, we increase the sphere slightly and compute the box $B_{i}$ to enclose the enlarged sphere. Computing the smallest enclosing sphere of $R_{i}$ takes randomized time $\mathbf{O}\left(m_{i}\right)$, see [24]. Given the boxes $B_{i}$, we find the overlapping pairs using the segment-tree streaming algorithm as described in [25]. Writing $b_{i}$ for the number of boxes that overlap $B_{i}$, we have a total of $b=\frac{1}{2} \sum_{i} b_{i}$ of overlapping pairs. The streaming algorithm takes time proportional to $n \log _{2}^{3} n+b$ to find them. For each pair of overlapping boxes, we check whether or not the critical region they enclose have a non-empty intersection. Standard computational geometry methods allow us to determine whether or not $R_{i}$ and $R_{j}$ intersect in time $\mathrm{O}\left(m_{i j} \log m_{i j}\right)$, where $m_{i j}=m_{i}^{2}+m_{j}^{2}$ [13].
STEP 0.3. The result of Steps 0.1 and 0.2 is a graph $R$. Its $n$ nodes are the critical regions, and its $q$ arcs are the pairs of critical regions with non-empty overlap. Writing $q=\frac{1}{2} \sum_{i} q_{i}$, where $q_{i}$ is the degree of the $i$-th node, we compute the cliques of size $2,3,4$ by checking all pairs and triplets of neighbors. Finding the cliques that include $R_{i}$ thus takes time $\mathrm{O}\left(\binom{q_{i}}{1}+\binom{q_{i}}{2}+\binom{q_{i}}{3}\right)$.

Analysis. The time for Step 0 is dominated by the requirement for Step 0.2 , which is some constant times $T_{\text {new }}=\sum_{i}\binom{q_{i}}{1}+\binom{q_{i}}{2}+\binom{q_{i}}{3}$. The time for Step 1 is some constant times $\left|S_{0}\right| \leq T_{\text {new }}$ and that for Step 2 is some constant times $T=\left|S_{1}\right| n \log _{2} n$. This adds up to some constant times $T_{\text {new }}+T$, as compared to $T_{\text {old }}+T$ for the algorithm in [4], where $T_{\text {old }}=\binom{n}{2}+\binom{n}{3}+\binom{n}{4}$. Any improvement thus hinges on two properties, namely that $T_{\text {old }}$ is significantly larger than $T_{\text {new }}$ as well as $T$. We now show that the first property holds under grossly simplifying assumptions, and we provide evidence in the next section that both properties hold for data we encounter in practice.

CAP ASSUMPTION. The critical regions are spherical caps, all of the same size, and their centers are uniformly distributed on $\mathbb{S}^{2}$.

Recall that the areas of the critical regions add up to the total absolute Gaussian curvature, $\sum_{i} \operatorname{Area}\left(R_{i}\right)=G(K)$. This sum is also half the area of the sphere times the average number of critical points of the height functions, $G(K)=2 \pi c_{\text {avg }}$. It follows the area of a single critical region is $\operatorname{Area}\left(R_{i}\right)=2 \pi c_{\mathrm{avg}} / n$, and because the cap is smaller than the flat disk of the same radius, its radius squared is $\rho^{2}>2 c_{\text {avg }} / n$. Two caps overlap if and only if the center of one is contained in the cap of radius $2 \rho$ around the center of the other. The area of the enlarged cap is less than four times $\operatorname{Area}\left(R_{i}\right)$. Hence the probability for a region $R_{j}$ to overlap $R_{i}$ is $\operatorname{Prob}\left[R_{i} \cap R_{j} \neq \emptyset\right] \leq 4 \operatorname{Area}\left(R_{i}\right) / 4 \pi=$ $2 c_{\text {avg }} / n$. Since expectations are additive even if the events are not independent, the expected number of $k$-tuplets of neighbors is $\left.\operatorname{Exp}\left[\begin{array}{c}q_{i} \\ k\end{array}\right)\right] \leq\binom{ n-1}{k} \operatorname{Area}\left(R_{i}\right)^{k} / \pi^{k} \leq$ $2^{k} c_{\mathrm{avg}}^{k} / k!$. Adding the expectations for $k=1,2,3$ and all $i$ gives

$$
\operatorname{Exp}\left[T_{\mathrm{new}}\right] \leq n \cdot\left(2 c_{\mathrm{avg}}+2 c_{\mathrm{avg}}^{2}+\frac{4}{3} c_{\mathrm{avg}}^{3}\right)
$$

Recall that $c_{\mathrm{avg}}=G(K) / 2 \pi$. It follows the average number of $k$-tuplets of critical regions overlapping a given one depends on the shape of the smooth surface and not on the size of the approximating triangulated surface. Similarly, the time for Step 0 depends on the shape and otherwise only linearly on the number of vertices in the triangulation.

## 4 Experiments

Input data. We use two types of triangulated surfaces approximating smooth models of biomolecular structures all listed in Table 1 Left. The first type is the molecular skin which uses hyperboloid and concave sphere patches to blend between the spheres that represent the atoms of a molecule [14]. An algorithm that constructs an approximating triangulated surface with guaranteed bounds on two- and three-dimensional angles is described in [7] and software written by Ho-lun Cheng is available at [3]. The second type is the molecular surfaces generated by Chimera [19]. The MSMS algorithm used in Chimera [20] constructs a triangulation of the solvent excluded surfaces initially computed by Connolly [11].

Critical point statistics. For each data set, we estimate the minimum, average, and maximum number of critical points of the height functions, which we sample at one thousand directions chosen from $\mathbb{S}^{2}$. The results are shown in Table 1 Middle. Comparing the estimated with the actual average, which we get using $c_{\mathrm{avg}}=G(K) / 2 \pi=$

Table 1. Left: the triangulated surfaces used in our computational experiments together with their numbers of vertices, edges, and triangles. Middle: estimated minimum, average, and maximum of the number of critical points of the height functions. Right: minimum, average, and maximum of the number of triangles needed to triangulate the critical regions. Last column: percentage of non-simple critical regions. Top: molecular skin surfaces. Bottom: molecular Chimera surfaces.

| id | name | $n$ | $m$ | $\ell$ | $c_{\text {min }}$ | $c_{\text {avg }}$ | $c_{\text {max }}$ | $\frac{c_{\text {avg }}}{n}$ | $r_{\text {min }} r_{\text {avg }} r_{\text {max }}$ | $\%$ |
| :--- | :--- | ---: | ---: | ---: | ---: | ---: | ---: | ---: | ---: | ---: |
| 0 | 1BRS-5to6 | 1,370 | 4,104 | 2,736 | 2 | 6.41 | 16 | 0.0047 | 23.99 | 8 |
| 1 | 1CLU-DBG | 3,149 | 9,441 | 6,294 | 2 | 13.50 | 44 | 0.0043 | 24.01 | 12 |

$\sum_{i} \operatorname{Area}\left(R_{i}\right) / 2 \pi$, we see that the error is small. For example, for data set 4 , the estimated $c_{\text {avg }}$ is 29.92 while the actual average is 29.94 . Since all our skin triangulations approximate a smooth surface to about the same accuracy, for different surfaces, the average number of critical points scales linearly with $n$. Indeed, $c_{\mathrm{avg}} / n$ is between 0.003 and 0.005 for all our skin data sets.

As mentioned earlier, each vertex of $K$ is critical for a region of directions, in fact two antipodal regions. Most of these regions are simple, that is, defined by a polygon without self-intersections. As shown in the last column in Table 1 the percentage of non-simple polygons is indeed rather small. Besides checking for self-intersections, we measure the complexity of a critical region by counting the triangles we need to triangulate it on the sphere. The minimum, average, and maximum of this number are given in the right half of Table 1

Intersection statistics. The following statistics were collected for the finer molecular skin surfaces only. Recall that we compute the pairs of intersecting critical regions in two steps, first finding the intersections among the bounding boxes and second among the critical regions. Table 2 Left gives the statistics for both.

Given a pair of intersecting boxes, we test whether or not the corresponding critical regions intersect by checking the overlap among the triangles in their triangulations. The average number of triangle-triangle checks is consistently between 11 and 12 , which justifies the use of this brute-force over a more sophisticated method.

Similar to the number of critical points, we expect that the average number of boxes intersecting a given box and the average number of critical regions intersecting a given critical region scale linearly with $n$. Indeed, $b_{\text {avg }} / n$ is between 0.04 and 0.07 and $q_{\text {avg }} / n$ is between 0.02 and 0.03 for all our skin data sets. The latter is about six times the average number of critical points; compare this with the factor two we got under the Cap Assumption. The observed relation between these two quantities is only about three times as loose, which is reasonable considering that real data necessarily violates the Cap Assumption to some extent (due to irregular shapes and different orientations of the critical regions). The new algorithm starts with $T_{\text {new }}$ tuplets. A back-of-the-

Table 2. Left: the minimum, average, and maximum number of boxes intersecting a given box; the minimum, average, and maximum number of critical regions intersection a given critical region. Middle: the number of cliques before and after the Projection Filter and the Persistence Filter. Right: dominant terms in the running time of the old and the new algorithms.

|  |  |  |  |  |  |  |  |  | $S_{0} \mid / 10^{3}$ | $\left\|S_{1}\right\|$ |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 0 | 12 | 94 | 207 | 0.069 | 9 | 40 | 97 | 0.029 | 1,608 | 2,373 | 15 | 24 | 3 |
| 1 | 27 | 204 | 626 | 0.06 | 11 | 82 | 250 | 0.026 | 32,119 | 20,521 | 410 | 508 | 749 |
| 2 | 52 | 236 | 556 | 0.056 | 20 | 92 | 201 | 0.022 | 43,572 | 17,175 | 1,356 | 720 | 82 |
| 3 | 95 | 243 | 859 | 0.040 | 29 | 134 | 330 | 0.022 | 198,023 | 56,797 | 5,820 | 3,327 | 4,368 |
| 4 | 99 | 423 | 1,276 | 0.054 | 35 | 160 | 543 | 0.021 | 433,116 | 94,300 | 15,411 | 7,354 | 9,508 |

envelope calculation suggests that $T_{\text {new }}$ is roughly $n\binom{q_{\text {avg }}}{3}$, which is roughly a factor of ten thousand smaller than $\binom{n}{4}$, independent of the value of $n$. We thus might expect the new algorithm runs about four orders of magnitude faster than the old one.

Running time. Recall that $S_{0}$ is the set of cliques of size 2,3 , or 4 in the intersection graph of the critical regions. The subset $S_{1} \subseteq S_{0}$ contains all cliques that pass the Projection Filter, and the subset $S_{2} \subseteq S_{1}$ contains all cliques that also pass the Persistence Filter. The sizes of the first two sets are given in the middle of Table 2 ,

Most relevant to the running time of the algorithms for computing elevation maxima is $S_{1}$. Indeed, both the old and the new algorithm start with sets of $2-, 3$-, and 4 -tuplets that contain the cliques in $S_{0}$ and much more. As shown in Table 2 on the right, the overestimate by the old algorithm is about ten thousand times that of the new algorithm. Furthermore, in the new algorithm, the time for Step 0 and Steps 1 and 2 is fairly balanced. This implies a speed-up of about four orders of magnitude, which is consistent with back-of-the-envelope calculation mentioned above.

Conclusions. The main result of this paper is a new algorithm for computing all elevation maxima of a triangulated surface in $\mathbb{R}^{3}$. We provide experimental evidence that for practical data, the new algorithm runs about four orders of magnitude faster than the old one. The improvement is achieved by making the running time dependent on the total absolute Gaussian curvature of the surface and to a lesser extent on the number of vertices in the approximating triangulation.

## References

1. Computational Geometry Algorithms Library, http://www.cgal.org
2. Banchoff, T.F.: Critical points and curvature for embedded polyhedral surfaces. Amer. Math. Monthly 77, 475-485 (1970)
3. The biogeometry web-pages, http://www.biogeometry.duke.edu
4. Agarwal, P.K., Edelsbrunner, H., Harer, J., Wang, Y.: Extreme elevation on a 2-manifold. Discrete Comput. Geom. 36, 553-572 (2006)
5. Alboul, L., Echeverria, G.: Polyhedral Gauss maps and curvature characterization of triangle meshes. LNCS, vol. 3605, pp. 14-33. Springer, Heidelberg (2005)
6. Cazals, F., Chazal, F., Lewiner, T.: Molecular shape analysis based upon the Morse-Smale complex and the Connolly function. In: Proc. 19th Ann. Sympos. Comput. Geom., pp. 351360 (2003)
7. Cheng, H.L., Dey, T.K., Edelsbrunner, H., Sullivan, J.: Dynamic skin triangulation. Discrete Comput. Geom. 25, 525-568 (2001)
8. Cohen-Steiner, D., Edelsbrunner, H., Harer, J.: Stability of persistence diagrams. Discrete Comput. Geom. 37, 103-120 (2007)
9. Cohen-Steiner, D., Edelsbrunner, H., Harer, J.: Extending persistence using Poincaré and Lefschetz duality. Found. Comput. Math. (to appear)
10. Cole-McLaughlin, K., Edelsbrunner, H., Harer, J., Natarajan, V., Pascucci, V.: Loops in Reeb graphs of 2-manifolds. Discrete Comput. Geom. 32, 231-244 (2004)
11. Connolly, M.L.: Analytic molecular surface calculation. J. Appl. Crystallogr. 6, 548-558 (1983)
12. Connolly, M.L.: Shape complementarity at the hemoglobin albl subunit interface. Biopolymers 25, 1229-1247 (1986)
13. de Berg, M., van Kreveld, M., Overmars, M., Schwarzkopf, O.: Computational Geometry Algorithms and Applications. Springer, Berlin (1997)
14. Edelsbrunner, H.: Deformable smooth surface design. Discrete Comput. Geom. 21, 87-115 (1999)
15. Edelsbrunner, H., Letscher, D., Zomorodian, A.: Topological persistence and simplification. Discrete Comput. Geom. 28, 511-533 (2002)
16. Georgiadis, L., Tarjan, R., Werneck, R.F.: Design of data structure for mergeable trees. In: Proc. 17th Ann. ACM-SIAM Sympos. Discrete Algorithm, pp. 394-403 (2006)
17. Morvan, J.: Generalized Curvatures. Springer, Heidelberg (2008)
18. Munkres, J.R.: Elements of Algebraic Topology. Addison-Wesley, Reading (1984)
19. Petterson, E.F., Goddard, T.D., Huang, C.C., Gouch, G.S., Greenblatt, D.M., Meng, E.C., Ferrin, T.E.: UCSF Chimera - a visualization system for exploratory research and analysis. J. Comput. Chem. 25, 1605-1612 (2004)
20. Sanner, M.F., Olson, A.J.: Reduced surface: an efficient way to compute molecular surfaces. Biopolymers 38, 305-320 (1996)
21. Santaló, L.: Integral geometry and geometric probability. Addison-Wesley, Reading (1976)
22. Cohen-Steiner, D., Morvan, J.: Second fundamental measure of geometric sets and local approximation of curvatures. J. Differential Geom. 74(3), 363-394 (2006)
23. Wang, Y., Agarwal, P.K., Brown, P., Edelsbrunner, H., Rudolph, J.: Course and reliable geometric alignment for protein docking. In: Proc. Pacific Sympos. Biocomputing 2005, pp. 64-75. World Scientific, Singapore (2005)
24. Welzl, E.: Smallest enclosing disks (balls and ellipsoids). In: Maurer, H.A. (ed.) New Results and New Trends in Computer Science. LNCS, vol. 555, pp. 359-370. Springer, Heidelberg (1991)
25. Zomorodian, A., Edelsbrunner, H.: Fast software for box intersections. Internat. J. Comput. Geom. Appl. 12, 143-172 (2002)

[^0]:    * This research is partially supported by the Defense Advanced Research Projects Agency (DARPA) under grants HR0011-05-1-0007 and HR0011-05-1-0057.

