# Real-time Visualization of Protein Empty Space with Varying Parameters

Ondřej Strnad, Vilém Šustr, Barbora Kozlíková and Jiří Sochor
*Faculty of Informatics*
*Masaryk university, Brno, Czech Republic*
*Email: xstrnad2@fi.muni.cz, xsustr@fi.muni.cz, xkozlik@fi.muni.cz, sochor@fi.muni.cz*

***Abstract*—Exploration of the empty space inside protein structures is playing a crucial role in protein engineering and drug design. This empty space inside proteins can be utilized for the design of protein mutations. The importance of this empty space is also based on its ability to accept a small ligand molecule which can react with the protein. The product of such a reaction can form the basis of new medications. Many algorithms enabling computation of these empty spaces, often marked as voids, have been published and their results were evaluated by protein engineers to confirm their chemical relevance. However, not all voids of a protein can be considered as a target point of ligand binding. Thus, the following examination and assessment of all voids must be performed. In this phase the visual representation of voids is very valuable and substantially decreases time of this evaluation phase.**

**In this paper, we introduce a novel algorithm for the visualization and further evaluation of these voids in real-time. This user-driven approach enables to compute and display empty space that satisfies the input parameters instantly. Basically, these parameters include setting of minimal desired width of the voids. The values of these parameters can be changed by the user anytime and the changes are immediately displayed and prepared for further exploration.**

***Keywords*-protein, empty space, void, visualization, real-time, cavity**

## I. Introduction

Long-term research in the area of protein analysis proved the importance of an empty space situated inside the macromolecular structure. This empty space can be further qualified according to various criteria and marked as a cavity, pocket, tunnel, channel, pore or other specific structure (see Fig. 1) . Inner cavities can serve as the destination for a small ligand molecule that can follow a pathway from the outside environment of the protein. The specific cavity accepting ligands is marked as an active site and in such cavity the chemical reaction between protein and ligand can take place. Products of such a reaction can serve as the basis for new drugs or various chemical compounds. A pocket can be defined as a hollow space on a protein's surface. This means that if the ligand is small enough the structure can be reached directly.

Channels and pores are specific pathways crossing the whole protein. They can be used for the transport of substrates, products, water molecules and other compounds through the protein. The distinction between channels and pores lies in their shape – a channel can have various curvatures whereas pores pass straight through the protein.

Our research in this field was concerned mainly with the detection of tunnels. These structures represent a path leading from a specific protein cavity (an active site) to the molecular surface. Thus for tunnel detection it is necessary to analyze and evaluate protein cavities to detect the active site in advance. Small substrate molecules entering the active site determine the minimal properties of computed tunnels, such as their width or curvature. Tunnel analysis and their further evaluation enhances the workflow of protein engineers or drug designers.

The derivation of empty space from the 3D structure of a protein's amino-acid sequence introduces a very complex task. Proteins with previously detected positioning in 3D space are stored in the well-known PDB database. This archive contains information about experimentally-determined structures of proteins, nucleic acids, and complex assemblies. Some of the PDB structures were thoroughly analyzed and active sites which were discovered were subsequently stored in the CSA (Catalytic Site Atlas) database. However, active sites of most of the structures still have not been revealed or published. This situation creates the necessity of using other semi-automated tools or even manually-detecting of the active site.

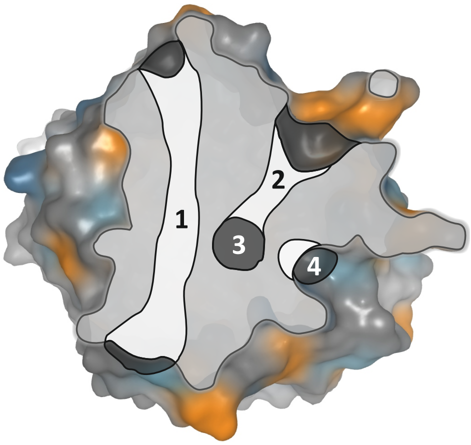


Figure 1. Illustration of a channel (1), a tunnel (2) leading from the active site (3) and a cavity (4).

ISBN: 978-1-61208-260-8

As mentioned above, the active site is placed inside a protein cavity. The first enhancement leads to the detection of all cavities inside the molecule. However, when operating with large protein complexes or even ribosomes, computation of all cavities in such molecule is very time and memory consuming. We introduce a novel method for detection and visualization of inner cavities focusing on minimizing the memory and time requirements. This technique is designed to operate in real-time, enabling users to interactively change the inner and outer size of the spherical probe utilized for empty space detection.

Without proper visualization it is complicated for biochemists to evaluate the computed cavity. Powerful visualization tool enabling not only displaying of detected voids but also allowing real-time alternations of parameters of detected voids is crucial for biochemists to be able to select the proper active site. When combining this method with other techniques, such as determination of partial charges, users can promptly recognize the possibility of occurrence of the desired chemical reaction. More specifically, when the surroundings of the cavity has neutral or small partial charge, this cavity probably will not be considered as an active site.

## II. Related work

Detection and classification of the empty space inside proteins has been in the scope of biochemists for the last decades, and number of algorithms have been published in this field. Although the aim of the method described in this article is to, above all, detect and visualize cavities, in this section we will introduce related research focusing generally on the detection of empty space. These techniques can be adapted for the computation of tunnels, channels etc.

Algorithms detecting empty space inside protein macromolecules share similar principle; they are all based on computational geometry using protein geometry (positions and radii of atoms) as the input. These algorithms can be divided into two groups according to their approach to space representation. The first group is based on a grid approach while the second one utilizes a Voronoi diagram and Delaunay triangulation. The main difference lies in their precision, speed and memory consumption.

The outer environment can be considered a void as well, thus the protein has to be encapsulated in a bounding object. The void detection process inside protein structures is highly influenced by the protein surface, which gives an overview of the protein's compactness. In most cases, the empty space is detected only within the volume that is defined by the surface. The construction of the molecular surface will be presented in the second part of the related work.

### A. Detection of empty space

*1) Grid method:* The entire protein is enclosed in an axis aligned bounding box subsequently sampled regularly to a voxel grid. Each vertex of the voxel grid is classified according to its collision with an atom. Non-colliding voxels form the empty space used for construction of cavities, tunnels and other structures. The quality of results is strongly influenced by the sampling density. Too sparse sampling can lead to a situation where all vertices of voxels are colliding with an atom and no empty space is detected. On the other hand, too dense a sampling causes an enormous increase in time and memory demands. The main advantage of this approach is its simplicity; the disadvantage, as already mentioned, follows from computational complexity $\mathcal{O}(n^3)$ with $n$ depending on the sampling density.

The grid approach was adopted for tunnel computation in CAVER 1.0 [1]. Another tool using the grid approach for computation of specific cavities (pores) inside proteins is called CHUNNEL [2]. Each voxel is marked according to its distance to the nearest atom. Onto this structure the Dijkstra algorithm is used and the tunnel with highest voxel values (the widest tunnel) is detected.

Kleywegt et al. [3] presented their grid approach applied to the detection of cavities. Their implementation is presented in the VOIDOO application. The first step of the algorithm maps the protein onto a 3D grid with a spacing between 0.5 and 1.0 Ångströms. Each point of the grid is noted by the zero value. Then, each grid point is processed and when the distance to the nearest atom is less than the sum of the atomic radius and probe radius, its value is set to one. This method is also known as the flood-fill algorithm. Finally, the points inside cavities still have a zero value, so they can be easily detected and their volume can be measured.

*2) Voronoi diagram and Delaunay triangulation:* Another approach to protein 3D space inspection is based on the Voronoi diagram (VD) and its dual structure - the Delaunay triangulation (DT). The benefit of this approach is the division of the space without any dependency on user defined variables, which overcomes the main disadvantage of the previous grid approach. A detailed description of VD construction can be found in [4]. The dual structure to VD, the Delaunay triangulation (tetrahedrization in the 3D case), can be constructed by connecting neighboring points sharing the Voronoi edge (see Fig. 3). Tetrahedra of the Delaunay tetrahedrization fulfills the condition that no point is presented inside the circumsphere of any tetrahedra.

Voronoi diagrams and Delaunay tetrahedra were utilized by various software tools for tunnel and channel computation, such as CAVER 2.0 [5], MolAxis [6] or MOLE [7].

Another approach to cavity detection using Delaunay triangulation and the alpha complex was implemented in the CAST application [8] (CASTp is its online version). It is also able to analytically measure the area and volume of cavities as well.

In [9], Voronoi diagrams were extended to the Additively weighted Voronoi diagrams (AVD). AVDs were originally designed for environments containing non-uniform objects. They can be used to geometrically analyze protein structures consisting of many atoms with different radii. Compared to traditional VDs, AVDs gain the more adequate space subdivision through the specification of weight $w$ attached to each site point. According to their weight values the respective points attract ($w > 1$) or repel ($w < 1$) the corresponding Voronoi edges. Resulting Voronoi edges have curvilinear shapes. AVD construction is more complex in comparison to traditional VD and thus the time and space complexity increases substantially. AVD were used in the protein visualization tool called Voroprot [10].

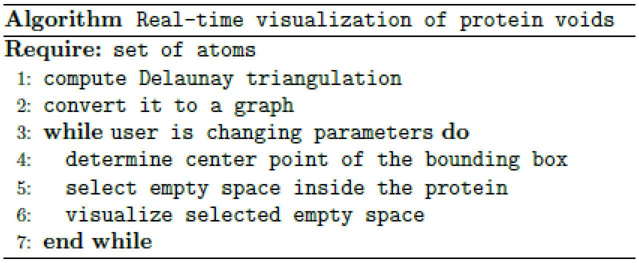
### B. Protein surface

Detection and visualization of surfaces play an important role not only in the case of detection of voids in proteins but in many other fields as well. Thus, surface detection has been in researchers' scope for decades, and many approaches have been proposed. Two main groups of existing algorithms employ either analytical or numerical approach.

*1) Analytical surface construction:* The input set contains objects that should be encapsulated by the surface. The analytical approach describes the surface using a set of mathematical equations. For protein exploration there are two basic analytical approaches to generation of surfaces. The reduced surface [11] is constructed by rolling a probe of specific radius over the protein outer boundary. Inwards facing parts of the probe surface combined with parts of atoms' surfaces on the boundary create the resulting solvent-accessible surface. The second approach is based on the alpha-shapes theory [12]. The main disadvantage of the analytical representation of the surface comes from its complexity. Thus its utilization on large datasets (e.g. macromolecular structures) cannot be performed in real-time or can even fail.

*2) Numerical surface construction:* The accuracy of numerically based algorithms is strongly dependent on initial user settings. The basic principle is the division of the scrutinized space into a uniform voxel grid. Each voxel is classified according to its intersection with objects in space. Subsequently, the marching cubes algorithm [13] can be utilized for the detection of the surface. The marching cubes method was designed primarily for a simple and fast construction of iso-surfaces in volume data sets. This approach is widely used e.g. in MRI or other medical applications.

*3) Visualization:* The combination of surface detection with its visualization is not only crucial for the exploration of protein shapes. Protein surfaces play important role in many chemical simulations and many methods for the visualization of proteins geometry and their chemical properties have been designed. They include both analytical or numerical representations, such as [14] or the LSMS algorithm [15].



```
Algorithm Real-time visualization of protein voids
Require: set of atoms
1: compute Delaunay triangulation
2: convert it to a graph
3: while user is changing parameters do
4:   determine center point of the bounding box
5:   select empty space inside the protein
6:   visualize selected empty space
7: end while
```

Figure 2. Overview of the algorithm.

## III. Real-time visualization of protein voids

In comparison with existing algorithms for highlighting empty voids, our approach does not require additional time for their re-computation when the input parameters change. The empty space corresponding to changed parameters is visualized immediately. In the rest of this section, the basic principle of our novel algorithm will be described. As noted above, the main aim of our approach is the real-time visualization of inner voids. Firstly, such voids must be computed. We utilize the standard Voronoi diagram, which omits the differences in radii of atoms (contrary to AVD approach) since our priority is the speed of the algorithm. From our experience, VD does not provide users with as precise results as AVD does, but the difference is not crucial for our purposes. The main difference between the results obtained by VD and AVD is in the exact representation of the surface of voids. However, the set of detected cavities is equal.

In the visualization phase, the algorithm is divided into five basic steps (see Fig. 2). Steps on lines 1 and 2 represent preprocessing and they are performed only once during the initialization phase. Steps on lines 4 to 6 (represented by subsections C to E) are iteratively repeated for any change of input parameters and are considered as the main contribution of this paper.

### A. Construction of Delaunay triangulation

**input:** set of atoms $A$
**output:** Delaunay triangulation $T$

The input set $A$ consists of all atoms of protein. Since we do not take into account the different radii, atomic centers were selected as representatives of atoms. The atomic centers then form the input set of points marked as $P$, which is subsequently processed. For the set $P$, the Delaunay triangulation $T$ is constructed using the QuickHull 4D algorithm described in [16].

The triangulation $T$ is afterward refined so that all tetrahedra intersecting the molecular surface of the protein are removed. In other words, all surface tetrahedra that are accessible from the outside by a probe with radius 2.8Å(double the van der Waals radius of oxygen) are removed from $T$

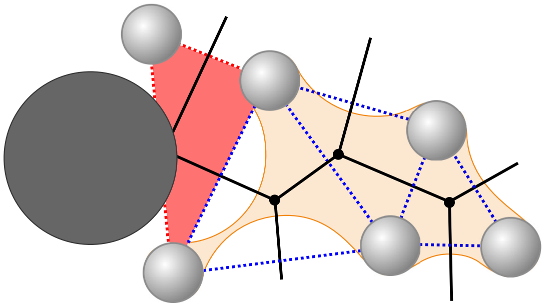


Figure 3. Tetrahedron (red) accessible by a probe (dark gray) is removed from the triangulation (blue dotted). The molecular surface defined by the probe is highlighted (orange).

(see Fig. 3). This ensures that the tunnel throat will not contain excessive boundary spheres.

*B. Construction of the graph $G$*

**input:** Delaunay triangulation $T$
**output:** evaluated graph $G$

For each tetrahedron $t_i \in T$ a node $N_i$ is inserted into a newly constructed graph $G$. An edge $e_{jk}$ connecting nodes $N_j$ and $N_k$ is added into $G$ if their referenced tetrahedra $t_j$ and $t_k$ share a face $f_{jk}$. For every edge $e_{jk} \in G$, we define its center point $c(e_{jk})$ and width $w(e_{jk})$ as follows. The center point $c(e_{jk})$ is defined as a point in $f_{jk}$ where sphere with maximal possible radius not intersecting any atom from $t_j$ or $t_k$ can be placed. The width $w(e_{jk})$ is then defined by the radius of such a sphere. The evaluation process is illustrated in Fig. 4.

*C. Selection of center point*

**input:** Delaunay triangulation $T$
**output:** center point $\mathbf{C}$

The algorithm was designed to operate with large macromolecules. In this case, computation and visualization of all inner voids usually leads to complex and ambiguous results, which the biochemist cannot properly explore, thanks to the huge amount of visualized data. In order to avoid this situation, we allow computing inner voids from a starting point $\mathbf{C}$ that represents the center of the bounding sphere. The empty space is then visualized only inside this bounding sphere - the area of interest. The point $\mathbf{C}$ set by the user can be determined in two ways. The user can enter the space coordinates of the point directly. In most cases the binding site is loaded from the CSA (Catalytic Site Atlas) database [17]. Once the center point $\mathbf{C}$ is set, it can be stored for further iterations of the algorithm.

*D. Selection of relevant edges*

**input:** graph $G$, point $\mathbf{C}$, distance $d$, parameter $w_{min}$
**output:** set of filtered edges $E$

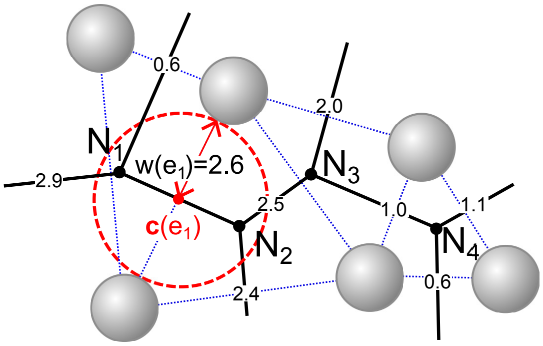


Figure 4. Illustration of a part of a graph G. Thick lines represent Voronoi edges. Every edge is evaluated by the value representing its distance to the nearest atom.

In this phase, the iteration process is started. The goal is to select a set $E$ of edges which satisfy the condition of thickness (driven by the parameter $w_{min}$ representing the minimal width of the edge) and proximity (parameter $d$ defining the bounding sphere radius). For remark, every edge $e_{jk}$ connecting two nodes $N_j$ and $N_k$ is evaluated by a width $w(e_{jk})$.

The set of filtered edges $E$ consists of all edges having the $w(e_{jk})$ greater or equal to $w_{min}$ and with the distance to $\mathbf{C}$ lower than $d$. More formally, let $G_E$ is the set of all edges from $G$. The set of filtered edges is then $E = \{e_{jk} \in G_E | dist(\mathbf{C}, c(e_{jk})) < d \wedge w_{min} \leq w(e_{jk})\}$.

*E. Visualization*

**input:** set of edges $E$, selected visualization method(s)

Firstly, the set $E$ has to be transformed into geometrical objects, which are possible to render. Every edge $e_{jk}$ is transformed into a sphere $s_{jk}$ with center in $c(e_{jk})$ and with radius equal to $w(e_{jk})$. The set $S$ of all such spheres is then prepared as an input for selected visualization method(s). For our case of protein visualization, we utilized two basic methods effectively describing the empty space inside macromolecules.

*1) Rendering of spheres:* represents the most intuitive visualization method, and also the fastest one (see Fig. 5). It simply displays all spheres of the set $S$. From the construction introduced above, all spheres fill the empty space inside the molecule and do not intersect with any atom. Using this method, the empty space is highlighted, but it looks distracting and for user it can sometimes be difficult to distinguish between an atom of the molecule and a sphere highlighting the empty space.

*2) Grid sampling:* enables users to visualize a continual surface of voids, which gives more intuitive and user friendly results. To construct such a surface, all spheres from the set $S$ are enclosed into an axis aligned bounding-box. This bounding-box is then regularly sampled with a user defined *density*. It is obvious that a higher density leads to a

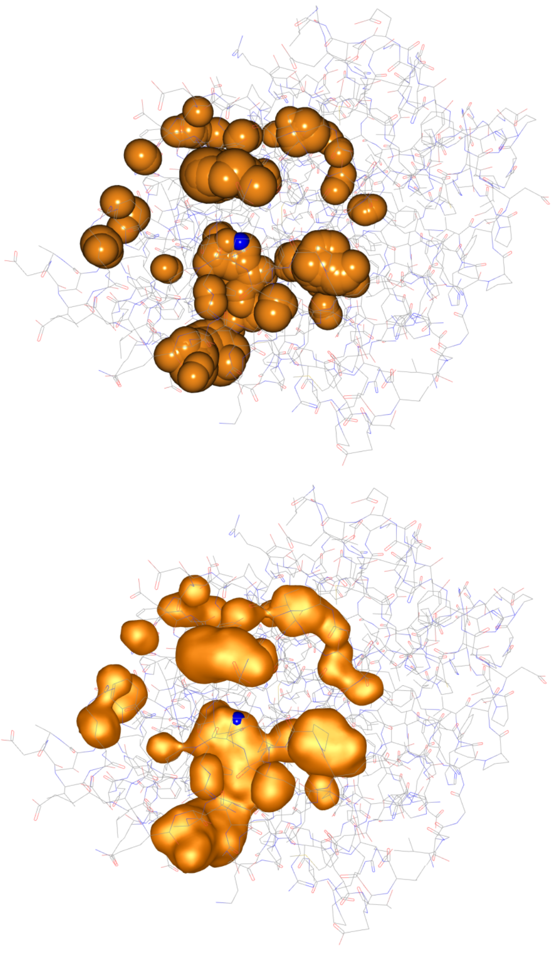


Figure 5. Empty space visualized as a set of spheres (top) or a surface (bottom).

more precise surface. On the other hand, the number of samples directly influences the memory and time complexity of the computation. We found out that for exploring of local neighborhood the empirically obtained $density = 200$ (i.e. grid 200x200x200) is optimal. Subsequently, every vertex of each cell in the grid is evaluated according to its intersection with any sphere from $S$. When all vertices are processed, the fully evaluated grid serves as the input for the marching cubes algorithm. For a notice, this algorithm operates with a predefined set of configurations, thus it its very straightforward and fast when constructing the resulting surface (see fig. 6).

## IV. Results and discussion

In this paper we presented a novel method for real-time visualization of empty space inside macromolecules which concentrates on user-driven evaluation of computed voids. The method is not limited by the size of the molecule (the number of atoms) as the encapsulation of displayed voids into a bounding sphere allows to restrict the amount of processed data respectively. The implementation does not demand any special hardware or software, the algorithm was implemented in 32-bit Java environment. The performance was tested on a common single-threaded 2.66GHz computer. Both rendering strategies, as well as various types of macromolecules (ranging from proteins to ribosomes) underwent this test. For the sphere rendering strategy and for the surface strategy where only the closer neighborhood ($|E| < 10000$) is visualized, the algorithm operates in real-time. On larger structures, where there is a necessity to process huge amount of edges, the interaction is not fluent but still operable. The examples of tested combinations are summarized in table I.

Table I

Performace of the algorithm meassured in FPS (Frames Per Seconds). Parameter $d$ represents a radius of a bounding sphere, $w(e)_{min}$ the minimal value of an edge.

| $d$ | $w(e)_{min}$ | **1CQW (≈3k atoms)** | | **1AON (≈60k atoms)** | |
|---|---|---|---|---|---|
| | | spheres | surface | spheres | surface |
| 15Å | 1.4Å | >100 | 26 | 73 | 16 |
| $max$ | 1.4Å | >100 | 19 | 28 | 2 |

For testing purposes haloalkane dehalogenase with approx. 3000k atoms (PDB ID 2HAD) and GroEL-GroES-(ADP)7 chaperonin complex with approx. 60k atoms (PDB ID 1AON) were chosen. To illustrate the real use of our algorithm, Fig. 5 visualizes the empty space in the 1CQW. The figure shows voids computed with user settings $w_{min} = 1.4$Å, $d = 15$Å and **C**$= 25; 95; 35$.

To evidence the relevance of empty space detected and visualized using our approach, we performed a comparison with results obtained by the well acknowledged CAVER algorithm. CAVER was designed for the detection of tunnels inside proteins and the results were thoroughly tested by the community of protein engineers [18]. Thus, to manifest the relevance of voids detected by our new approach, the computed voids must contain all detected structures such as tunnels or cavities. We verified that tunnels detected by the CAVER algorithm lead through the empty space highlighted by our method (see Fig. 6). All visualized tunnels can be subsequently compared with protein empty spaces in their neighbourhood simply by changing few visualization parameters.

## V. Future

The first extension of our implementation should lead to the parallelization of the marching cubes algorithm on the modern graphic cards [19]. Such implementation would substantially increase the performance of the rendering phase. We expect to be able to apply the surface method on large macromolecules in real-time.

## Acknowledgment

This work was supported by The Grant Agency of the Czech Republic, Contract No. P202/10/1435.

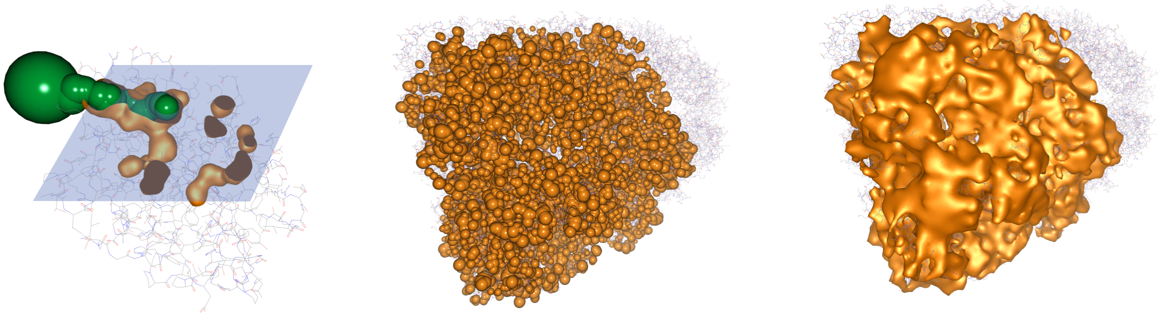


Figure 6. Left: Tunnel (green) detected by the CAVER algorithm lies inside the surface (bronze). The molecule is cut by a clipping plane (blue). Rendering of large protein structure, 1AON (approx. 60k atoms) by spheres (middle), surface rendering mode (right).

## REFERENCES

[1] M. Petřek, M. Otyepka, P. Banáš, P. Košinová, J. Koča, and J. Damborský, "Caver: A new tool to explore routes from protein clefts, pockets and cavities," *BMC Bioinformatics*, vol. 7, p. 316, 2006.

[2] R. G. Coleman and K. A. Sharp, "Finding and characterizing tunnels in macromolecules with application to ion channels and pores," *Biophysical Journal*, vol. 96, no. 2, pp. 632 – 645, 2009.

[3] G. J. Kleywegt and T. A. Jones, "Detection, delineation, measurement and display of cavities in macromolecular structures," *Acta Crystallographica Section D*, vol. 50, no. 2, pp. 178–185, Mar 1994. [Online]. Available: http://dx.doi.org/10.1107/S0907444993011333

[4] F. Aurenhammer, "Voronoi diagrams a survey of a fundamental geometric data structure," *ACM Comput. Surv.*, vol. 23, no. 3, pp. 345–405, sep 1991. [Online]. Available: http://doi.acm.org/10.1145/116873.116880

[5] P. Medek, P. Beneš, and J. Sochor, "Computation of tunnels in protein molecules using delaunay triangulation," *Journal of WSCG*, vol. 15(1-3), pp. 107–114, 2007.

[6] E. Yaffe, D. Fishelovitch, H. J. Wolfson, D. Halperin, and R. Nussinov, "MolAxis: efficient and accurate identification of channels in macromolecules." *Proteins*, vol. 73, no. 1, pp. 72–86, oct 2008. [Online]. Available: http://dx.doi.org/10.1002/prot.22052

[7] M. Petřek, P. Košinová, J. Koča, and M. Otyepka, "Mole: a voronoi diagram-based explorer of molecular channels, pores, and tunnels," *Structure*, vol. 15, no. 11, pp. 1357 – 1363, 2007.

[8] J. Liang, H. Edelsbrunner, and C. Woodward, "Anatomy of protein pockets and cavities: Measurement of binding site geometry and implications for ligand design," *Protein science : a publication of the Protein Society*, vol. 7, pp. 1884–1897, sep 1998.

[9] H. Edelsbrunner, *Algorithms in combinatorial geometry*. New York, NY, USA: Springer-Verlag New York, Inc., 1987.

[10] K. Olechnovič, M. Margelevičius, and v. Venclovas, "Voroprot," *Bioinformatics*, vol. 27, no. 5, pp. 723–724, mar 2011. [Online]. Available: http://dx.doi.org/10.1093/bioinformatics/btq720

[11] M. F. Sanner, A. J. Olson, and J. C. Spehner, "Reduced surface: an efficient way to compute molecular surfaces," *Biopolymers*, vol. 38, pp. 305–320, Mar 1996.

[12] H. Edelsbrunner, D. G. Kirkpatrick, and R. Seidel, "On the shape of a set of points in the plane," *IEEE Transactions on Information Theory*, vol. 29, pp. 551–559, jul 1983.

[13] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," *SIGGRAPH Comput. Graph.*, vol. 21, no. 4, pp. 163–169, Aug. 1987. [Online]. Available: http://doi.acm.org/10.1145/37402.37422

[14] M. Totrov and R. Abagyan, "The contour-buildup algorithm to calculate the analytical molecular surface," *J Struct Biol*, no. 1, pp. 138–43, 1995.

[15] T. Can, C. Chen, and Y. Wang, "Efficient molecular surface generation using level-set methods," *Journal of Molecular Graphics & Modelling*, vol. 25, pp. 442–454, 2006.

[16] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Trans. Math. Softw.*, vol. 22, no. 4, pp. 469–483, dec 1996. [Online]. Available: http://doi.acm.org/10.1145/235815.235821

[17] C. T. Porter, G. J. Bartlett, and J. M. Thornton, "The catalytic site atlas: a resource of catalytic sites and residues identified in enzymes using structural data," *Nucleic Acids Research*, vol. 32, no. Database-Issue, pp. 129–133, 2004.

[18] E. Chovancová, A. Pavelka, P. Beneš, O. Strnad, J. Brezovský, B. Kozlíková, A. Gora, V. Šustr, M. Klvaňa, P. Medek, L. Biedermannová, J. Sochor, and J. Damborský, "Caver 3.0: A tool for the analysis of transport pathways in dynamic protein structures," *PLoS Comput Biol*, vol. 8, no. 10, p. e1002708, 2012. [Online]. Available: http://dx.doi.org/10.1371/journal.pcbi.1002708

[19] C. Dyken, G. Ziegler, C. Theobalt, and H.-P. Seidel, "High-speed marching cubes using histopyramids," *Computer Graphics Forum*, vol. 27, no. 8, pp. 2028–2039, 2008. [Online]. Available: http://dx.doi.org/10.1111/j.1467-8659.2008.01182.x