Detection and visualisation of molecular surfaces on large macromolecules

Matus Kotry[*]
Supervised by: Jiri Sochor[†]

Faculty of Informatics
Masaryk University
Brno / Czech Republic

Abstract

Solid molecular surface is a simplified and clear representation of molecule, when you need to examine interaction of molecule and its surroundings. Therefore, the ability to compute and visualize molecular surface is expected from every modern software for molecular visualization. The main limitation of current applications enabling the computation of molecular surface lies in processing of large macromolecules, such as ribosomes. Traditional analytic solutions are not suitable for computation of surface for such large structures so another approach has to be used in such a case.

This paper describes how the molecular surface is detected and visualized using the LSMS grid algorithm [2]. Our implementation optimized the original algorithm and equipped it with advanced features which enhance resulting appearance of the surface. The algorithm was utilized to detect and visualize cavities inside molecules. The implementation was integrated to the CAVER Viewer application [3] for analysis and visualization of proteins.

Keywords: molecular surface, molecule, protein, ribosome, cavity, CAVER Viewer

1 Introduction

Proteins play an irreplaceable role in all living cells so the understanding of their functionality is substantial in many research fields, such as protein engineering, drug design, cosmetics, food industry and others. Analysis of proteins helps to understand their complex structure and ability to react with other molecules (called ligands) which changes the chemical properties of both proteins and ligands.

The rapid growth of the field of protein analysis in the last decades caused also the birth of many software tools enhancing the process. They concentrate on the algorithms performing various analyses on proteins as well as on the visualization of results. While the algorithms produce large amount of raw output data, the visualization tools concentrate on their processing and displaying relevant data in proper context. Modern visualization tools display the three-dimensional view of the protein structure. To provide users with more views on the output data, the tools offer a set of displaying methods designed for biochemists. Each of the methods concentrates on a specific level of abstraction in order to display only the most important information in given situation. Users can visualize for example all atoms of a molecule, all atoms with bonding, or secondary structures (see Figure 1). One of the most common representations is the visualization of molecular surface (Figure 1). The technique is often combined with other methods to provide the better overview.

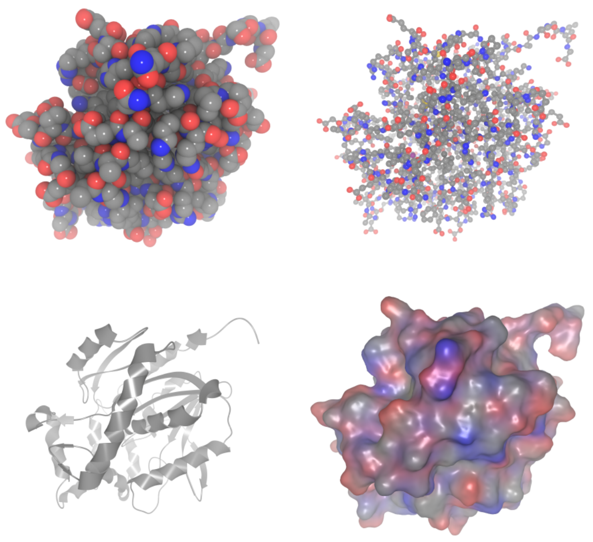
Figure 1: A protein visualized using different methods: top left - Van der Waals spheres, top right - Balls and Sticks, bottom left - Secondary structures, bottom right - molecular surface.

Computation and visualization of molecular surface is a big challenge for all the visualization tools, especially in case of large macromolecules. Due to the high time complexity we have to select a suitable tradeoff between the preciseness of the surface and the computational complexity. The problem attracted researchers for many years and

[*]xkotry@fi.muni.cz
[†]sochor@fi.muni.cz

various solutions were proposed. In this paper we present a solution which produces high-quality molecular surfaces computed in a few seconds even for large macromolecules. Our approach combines the existing LSMS technique ([2]) with the post-processing phase where the resulting surface is smoothed and colored.

2 Background

As mentioned before, the generation of a molecular surface present a complex task. The surface represents a boundary between the inner part of the protein and the outside environment called solvent. In this section we first concentrate on the definition of molecular surface and then present the well-known approaches to its computation.

2.1 Definition of molecular surface

Proteins are influenced by its environment, namely by the outer solvent. This causes permanent movements of protein atoms which are trying to maintain the state with minimal energy of the molecule. Thanks to these movements it is impossible to detect the molecular surface exactly and also definitions of the surface differ in literature. One of the most commonly used definitions operates with the protein represented by spheres with radii equal to corresponding van der Waals radii of atoms. In this representation, when two spheres overlap, the corresponding atoms are bonded. The union of these spheres is marked as van der Waals volume. The surface of this volume is called van der Waals surface (see Figure 2).

Definitions of other surfaces are derived from the concept of a probe rolling over the boundary atoms of the molecule. Union of centers of all possible position of the probe with a fixed radius touching the boundary atoms defines the so-called Solvent Accessible Surface (or Connoly surface) (see Figure 2). The same probe defines also the Solvent Excluded Surface (Molecular Surface) which touches the surface of boundary atoms (see Figure 2).

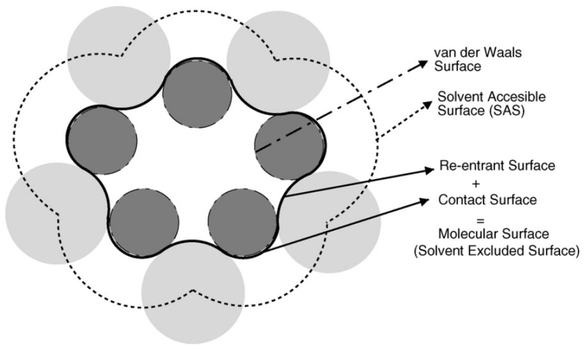
van der Waals Surface
Solvent Accesible Surface (SAS)
Re-entrant Surface
+
Contact Surface
=
Molecular Surface (Solvent Excluded Surface)

Figure 2: Various definitions of molecular surface (taken from [2]).

2.2 Related work

In this section we present existing algorithms for molecular surface computation and visualization. Generally, these algorithms can be divided into two groups. The first group contains algorithms which compute the analytical surface, the second group is formed by the algorithms computing the surface numerically.

2.3 Analytical approaches

The main goal of these algorithms is to compute an exact representation corresponding to the surface definition. The well known representatives of this group are Alpha Shapes [5] and Reduced Surfaces [11].

The Alpha Shapes algorithm is based on the generalization of the convex hull of a set of points. The definition is independent of the dimensionality of the explored space but in the case of molecular surfaces the three-dimensional space is taken into account. The accuracy of the surface is dependent on the value of alpha, which reaches from 0 to infinity. Value 0 represents the most precise surface and infinity stands for the convex hull of the molecule. The proper description of this complex algorithm can be found in [5].

Solvent Accessible and Solvent Excluded surface can also be computed using the Reduced Surface approach. The main advantage of this algorithm is its complexity which is in the $O(n * log n)$ set. This approach in the first phase constructs the so called reduced surface which is generated by rolling a sphere over the boundary atoms. In the second phase the analytical representation of the Solvent Excluded Surface is constructed. This phase can lead to the occurrence of self-intersections which are eliminated in the third phase. In the last phase, the triangulation of the final surface is generated. The resulting reduced surface corresponds to the surface generated using Alpha Shapes, when the alpha value is set to the radius of the probe used in Reduced Surfaces.

Another representatives of this group of algorithms can be Contour-Buildup Algorithm [13] or Power Diagram Algorithm [1].

The main problem of the analytical approaches is their dependence on the size and shape of the protein molecule. In the case of large macromolecules, such as ribosomes (hundreds of thousands of atoms), the computational time of these methods increases dramatically and the computation can even fail due to high memory requirements.

2.4 Numerical approaches

These algorithms overcome the problem of generating molecular surface for large macromolecules because it is independent of their size. Their principle is based on approximating the molecular surface, mostly by discretization of the space using voxel grid. The preciseness of the results depends on the size of voxels of the grid (grid resolution). This resolution also influences the overall computational time. In this case, the atom density does not play the important role because processing of voxels in the grid is independent of the density.

One of the first algorithms, which represent this group, is the Grid algorithm [7]. This algorithm places a set of probes onto the boundary voxels of the grid and then iteratively shifts these probes closer to the molecule (see Figure 3). The iteration for a probe ends when it collides with some atom. When all probes collide with some atoms, the algorithm is finished. Then the resulting surface is defined by positions of probes. The main limitation of this algorithm lies in its inability to detect some non-convex parts of the surface as illustrated in Figure 3. The blue line depicts the surface detected by this algorithm. The red area should be excluded from the volume of the molecule. So the molecular surface then should follow the boundary of this red area.

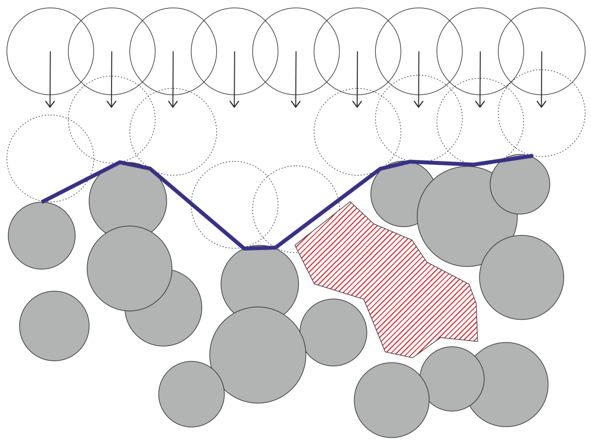

Figure 3: Principle of Grid algorithm.

The idea of approximation of the molecular space by three-dimensional grid was also adopted by authors of the LSMS algorithm [2]. Thanks to its preciseness and applicability to large macromolecules, this approach forms a basis of our solution presented in this paper. In the next sections we briefly describe the basic principles of the LSMS algorithm and then we concentrate on the key part of the paper - the post-processing phase which substantially improves the resulting surface obtained by LSMS.

3 Description of LSMS algorithm

LSMS stands for Level Set method for molecular surface generation and represents an advanced numerical algorithm. Similarly to the Grid algorithm, it is based on the discretization of space using voxel grid, however the surface itself is detected using a method called Level Set evolution.

Let $f(x,y)$ be function defined on discrete domain. Then level set of level c is a set of all points $P(x,y)$ where $f(x,y) = c$. Level set evolution is the process of detecting level set of level c, assuming we already know the level set of previous $c-1$ level.

Now we assume that the $f(x,y)$ function represents the distance map, where the value of each point is equal to its distance from edge of the nearest atom (considering that atoms are represented by spheres of van der Waals radii). If the point lies inside an atom, its value will be negative. Values in atom centers are then equal to $-r_w$, where r_w represents van der Waals radius of given atom, so we can use them to initialize the algorithm. From this knowledge we can derive other level sets.

Level set of level 0 in this function equals to the van der Waals surface. Level set of level r_p, where r_p is probe radius, equals to the Solvent Accessible Surface. Now let's define new distance map $g(x,y)$ where all points lying on and outside the Solvent Accessible Surface derived in previous step are initialized to zero (in other words, all points not visited during the previous level set evolution). Level set of level r_p then eqauls to Solvent-excluded surface, which is the desired output of the method.

As you can see, all we need to compute the final Solvent-excluded surface are two executions of the Level Set evolution. Level Set evolution itself is implemented by the method called Fast Marching. Output of the method are binary voxel data. There are more options, what can we consider as output. First option is, to output only the edge voxels of the surface, or in other words, the voxel corresponding of Level Set r_p of function $g(x,y)$.

Other option is to output molecule as a solid volume, from which Solvent-excluded surface can by extracted as a boundary between this volume and its surroundings. This representation is much more appropriate if we want to visualize the surface using an algorithm such as Marching Cubes. To get this representation we have to mark all voxels visited in the first Level Set evolution and then unmark all those visited in the second Level Set evolution.

3.1 Visualization of isosurface

LSMS generates the output in the form of voxel data which have to be further processed in order to generate the surface suitable for visualization. Computer graphics offers various algorithms for voxel data visualization. One of the most common methods is the Marching Cubes algorithm [10]. It generates the triangle mesh according to configurations of voxels in a grid (all possible combinations of intersections between voxel and objects in the grid space). In 3D space, there are 256 possible configurations for a cube consisting of 8 voxels.

In the next phase, the resulting triangle mesh has to be equipped with normal vectors for each vertex. This step is crucial mainly for computation of smooth lighting of the surface. The resulting surface generated by the LSMS algorithm can be seen in Figure 4. Here also the unpleasant bumpiness of the surface can be observed. This artifact as well as the absence of coloring surface according to chemical elements of corresponding atoms led us to introducing further improvements which overcome these problems.

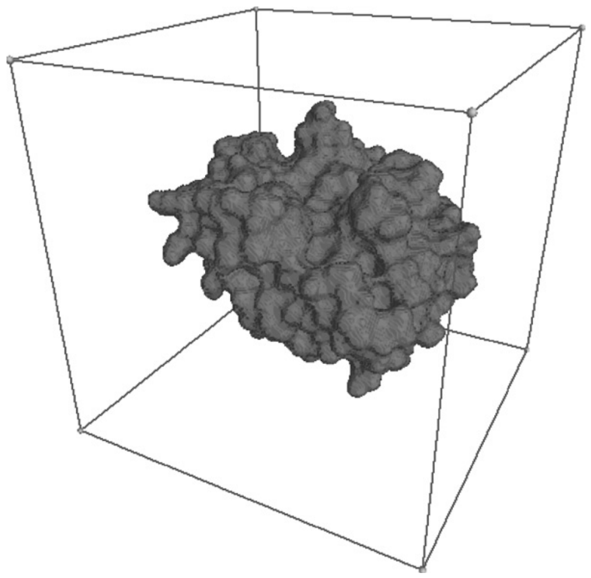

Figure 4: Resulting surface obtained by LSMS algorithm. Boundaries of the grid are also shown (taken from [2]).

4 Post-processing phase

This part forms the core of our improvement in comparison with results obtained by the LSMS algorithm and application of the Marching Cubes approach. The improvement basically consists of two parts - smoothing the surface generated by Marching Cubes and coloring the surface according to various characteristics.

4.1 Smoothing

As mentioned above, the isosurface generated by processing of LSMS voxels by Marching Cubes algorithms produces unpleasant artifacts. The resulting surface is not smooth, it resembles contour-lines on 3D maps. This is caused by the voxelization of the space. The original solution described in [2] does not solve this problem. One of possible solutions can be an interpolation of the vertex position. However, this approach is applicable only when working with greyscale voxel object. Moreover, its application to the resulting molecular surface has no effect. In this case, we have to come with more sophisticated solution.

Our solution is inspired by methods common in the field of digital image processing. In the space of 2D images, the desired quality can be reached by applying appropriate filtering method. We adopt this solution to the triangular mesh in 3D space.

The most suitable filter for our case is the Gaussian blur. For simplification, we describe its principle in 2D image space and then explain the extension to three-dimensional space. The 2D digital image can be considered as a regular graph where each pixel (node) has 4 neighbors (edges). Then the Gaussian blur can be applied by using the convolution kernel shown in Figure 5 (left part).

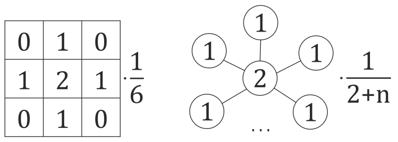

Figure 5: Left: convolution kernel of Gaussian blur for 2D digital image. Right: the same kernel for irregular graph.

When extending this principle to 3D space, we come out from the idea that the triangular mesh can be also considered as a graph which is irregular (variable number of edges corresponds to each vertex). Thanks to the symmetry of the 2D Gauss convolution kernel, the extension to three-dimensional space is straightforward. Each vertex v is multiplied by 2 and its neighboring vertices are multiplied by 1 (see Figure 5, right part). These coefficients are then normalized according to the number of neighbors of v. All resulting values are summed and this sum is stored as a new value of vertex v.

This process is applied on all vertices of the triangular mesh as well as on their normals to produce smooth lighting. The filter has very similar effects as in case of applying on 2D digital images - it preserves the overall shape of the object but smooths fine details such as the contour-line effect. Even smoother surface can be reached by repetitive application of this procedure. Figure 6 shows the original surface without any filtering. The smoothed surface after 3 applications of Gauss blur is depicted in Figure 7.

More precise result would be achieved if coefficients applied to neighbor vertices would not be constant but derived from Gaussian distribution formula according to their distance from the center vertex. However, differences between lengths of the edges in our mesh are negligibly small, because the mesh is generated using marching cubes algorithm. Marching cubes algorithm generates only polygons of certain shapes and sizes and therefore it can be easily proven that length of the longest edge generated by this algorithm is only twice as large as the shortest edge. In this case the errors resulting from usage of constant coefficients for all neighbors are visually unrecognizable in the result. This filter also shrinks the mesh slightly but the consequences of this shrinking are negligible for our purposes.

4.2 Coloring

The surfaces generated by the original LSMS algorithm were colored uniformly using one color with different intenzity only. This color scheme is quite limiting because it does not reveal many interesting features of surface related to the atoms forming the surface. For biochemists,

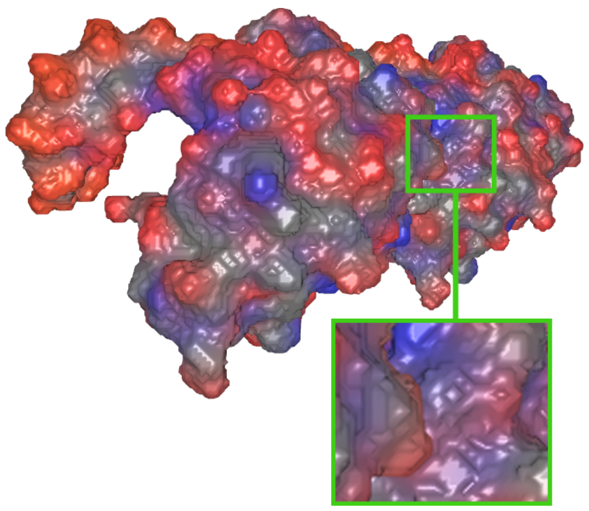

Figure 6: Original surface with no Gaussian blur. It corresponds to the surface obtained by LSMS.

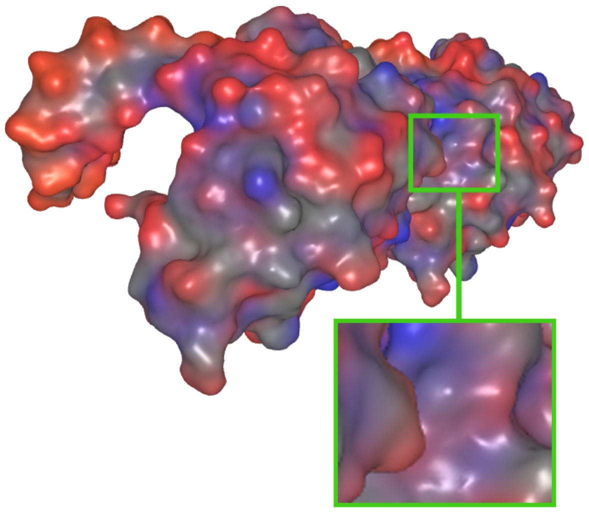

Figure 7: Surface obtained by three applications of Gaussian blur.

the visualization of such features can have an outstanding benefit. The surface can be colored according to physical and chemical properties of corresponding atoms (shown in Figures 6 and 7), their partial charges, correspondence to amino-acids, hydrophobicity of amino-acids of given atoms and others.

The color of the surface is computed for each vertex of the triangular mesh as a weighted sum of colors of atoms in the neighborhood of the vertex. For simplicity, we take into account only atoms in the distance of 5 Å(Ångström) from the given vertex. This constant is derived from the following consideration: all atoms present in proteins have the value of van der Waals radius smaller than 3 Åand the maximal size of the probe used for the surface construction in the analytical solutions is 2 Å. These two assumptions ensure that for each vertex on the surface at least one atom exists the center of which is closer than 5 Å.

Our function of computation of color in the vertex v was designed as follows:

$$c_v = \frac{\sum_{a \in A; d(a,v) \leq 5} w(a,v) c_a}{\sum_{a \in A; d(a,v) \leq 5} w(a,v)}$$

where

- c_v is the color of the vertex v
- c_a is the color of the atom a (user-defined according to the chosen color scheme)
- A is the set of all atoms of the protein structure
- $d(a,v)$ is the distance between the center of the atom a and the vertex v
- $w(a,v)$ is the weight defined as

$$w(a,v) = (5 - d(a,v))^3$$

A naive algorithm for coloring the surface processes all vertices and for each vertex checks the color of all atoms of the protein. In this case the time complexity is in $O(v * a)$ set, where v is the number of vertices and a stands for the number of atoms. It is obvious that this solution is highly dependent on the number of atoms in the protein and this is in contradiction with the principle of the LSMS algorithm. To overcome this problem, we introduced the following optimization.

The space occupied by the protein structure is covered by the regular searching space grid where the size of each cell is fixed to 5 Ångströms. Of course, there are plenty of more sophisticated space division structures (such as k-D tree or BSP tree), but for our purpose the regular grid is the most suitable solution because of guaranteed regular distribution of atoms within a molecule. Each cell of the grid contains the list of atoms whose centers lie inside the cell. This sorting of atoms has linear complexity with respect to the number of atoms. So when we return to the naive approach described above, for all vertices we now traverse only the atoms in the corresponding cell and its neighboring cells. The number of neighboring cells is 26 which together with the corresponding cell form the space cube with $15x15x15$ Ångströms. Thanks to limitations of mutual distances between atoms given by their chemical properties, the number of atoms which can be positioned in this cube is limited by a constant value k. Then the complexity of the computation of coloring the surface is in the $O(v * k) \approx O(v)$ set.

5 Other features

5.1 Cavities

One of the greatest advantages of the LSMS algorithm is its applicability to problem of detecting inner cavities of protein structures. Cavities represent an empty space present inside the spatial structure of proteins. Their presence is tightly related to protein density - dense proteins have less and smaller cavities than the sparse ones. Cavities play a crucial role when the protein reacts with other molecules - ligands. From the geometrical point of view, a cavity is defined as inner void from where a probe positioned inside cannot access the molecular surface (see Figure 8).

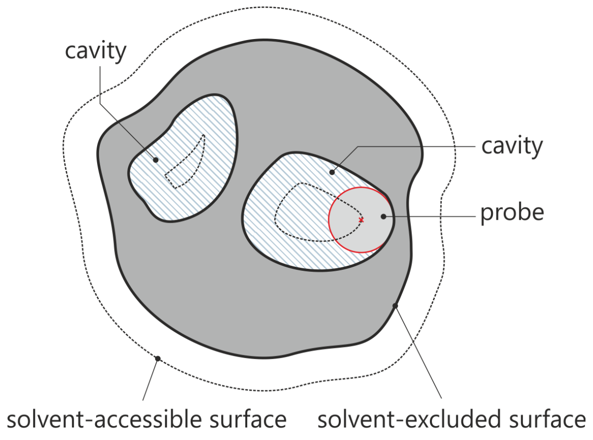

Figure 8: Illustration of cavity.

When adopting the LSMS algorithm for computation of cavities, it is necessary to distinguish between surface of the molecule and surface of inner cavities. To reach this, we add another Level Set evolution proceeding from the outer boundary of the voxel grid to its center. This step is performed before the final Level Set evolution of the original algorithm is launched. The resulting cavities detected by this approach can be seen in Figure 9.

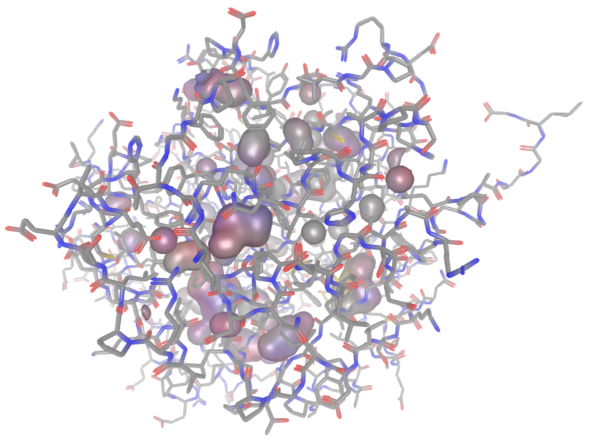

Figure 9: Cavities computed by our approach containing the post-processing phase.

5.2 Transparency

For biochemists it is very valuable to observe the overall shape of the molecule represented by its surface along with its inner structure. This can be accomplished by making the molecular surface transparent. The resulting surface is illustrated in Figure 10. To prevent side effects of transparency caused by wrong ordering of transparent polygons in Z axis (especially when more surfaces are present in the scene), only front facing polygons are actually made transparent. Back faces are blended with the color of the scene background, no matter what objects are behind them.

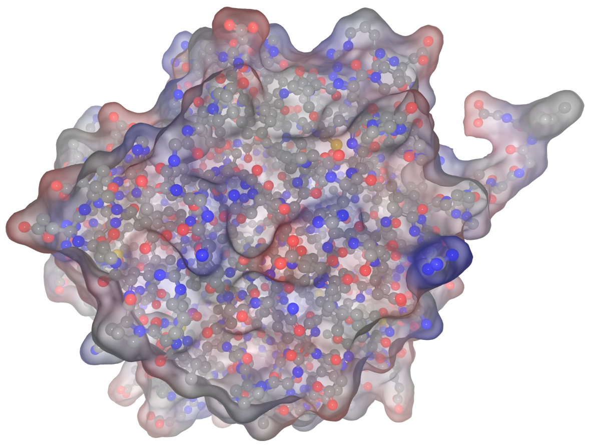

Figure 10: Transparent molecular surface.

6 Results

Our implementation of LSMS algorithm enhanced with the smoothing and coloring phase, was integrated into the CAVER Viewer application [4] for analysis and visualization of protein tunnels. This application provides users with an intuitive interface for computation and manipulation with molecular surfaces. Users are also able to set the coloring scheme according to the predefined options as well as customize all colors. Results of applying different coloring schemes on the protein surface computed in CAVER Viewer can be seen in Figures 11 and 12.

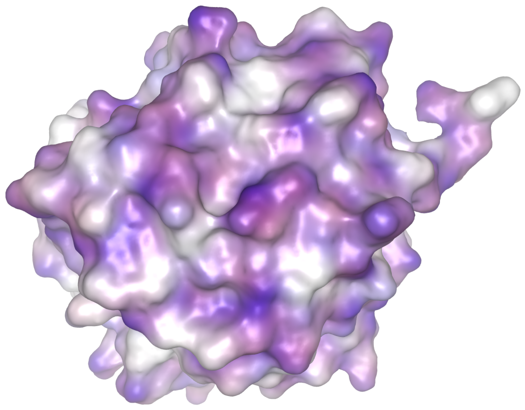

Figure 11: Molecular surface colored according to partial charges of atoms.

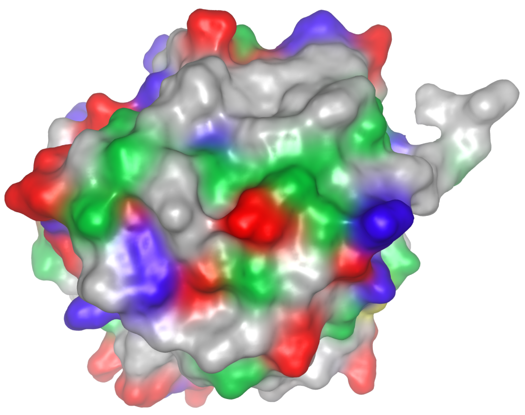

Figure 12: Molecular surface colored according to residues of corresponding atoms.

The main goal of our improvement was to provide biochemists with a visually appealing molecular surface which could be computed on molecules of arbitrary size. We compared the solution with other existing applications for protein visualization, such as PyMol [12], VMD [8] or YASARA [9]. Table 1 shows times spent on computation of molecular surfaces for three molecules of different size. First of them, haloalkane dehalogenase (1CQW code [6]), contains 2 754 atoms. Second structure, Groel/Groes complex (1AON code) consists of 58 874 atoms (Figure 13). The largest structure, 70S ribosome, contains 146 558 atoms (Figure 14). Our algorithm was launched with two different initial settings of the grid resolution - 128^3 voxels and the finer grid with 256^3 voxels. All applications were tested on the same computer, with Intel Core i5 M430 processor, nVidia GeForce GT 330M GPU and 4GB RAM. Dash in the table indicates that tested application was unable to provide any result for the given structure on this computer.

Compared to other approaches, our method is able to compute molecular surface for larger macromolecules, such as ribosomes. Moreover, the computational time is still reasonable.

6.1 Conclusion

The technique utilizes the LSMS algorithm which is equipped with additional smoothing and coloring phase to produce more plausible surface. To enhance the quality of protein analysis, the transparent surface was introduced. The method is applicable to ribosomes using commonly available desktop computers.

References

[1] Jr. A. Varshney, F. P. Brooks, and W. V. Wright. Computing smooth molecular surfaces. *IEEE Computer Graphics & Applications, 15, 19-25*, 1994.

[2] T. Can, C. I. Chen, and Y. F. Wang. Efficient molecular surface generation using level-set methods. *Journal of Molecular Graphics and Modelling, 25, 442454*, 2006.

[3] E. Chovancova, A. Pavelka, P. Benes, O. Strnad, J. Brezovsky, B. Kozlikova, A. Gora, V. Sustr, M. Klvana, P. Medek, L. Biedermannova, J. Sochor, and J. Damborsky. Caver 3.0: A tool for the analysis of transport pathways in dynamic protein structures. *PLoS Computational Biology 8: e1002708*, 2012.

[4] E. Chovancova, A. Pavelka, P. Benes, O. Strnad, J. Brezovsky, B. Kozlikova, A. W. Gora, V. Sustr, M. Klvana, P. Medek, L. Biedermannova, J. Damborsky, and J. Sochor. Caver 3.0. Software, 2011.

[5] H. Edelsbrunner and E. P. Mücke. Three-dimensional alpha shapes. *ACM Transactions on Graphics, 13(1)*, 1994.

[6] F.C.Bernstein, T.F.Koetzle, G.J.Williams, E.E.Meyer Jr., M.D.Brice, J.R.Rodgers, O.Kennard, T.Shimanouchi, and M.Tasumi. The protein data bank: A computer-based archival file for macromolecular structures. *J. of. Mol. Biol., 112:535*, 1977.

[7] J. Greer and B. Bush. Macromolecular shape and surface maps by solvent exclusion. *Proceedings of the National Academy of Sciences of the United States of America, 75, 303307*, 1978.

[8] W. Humphrey, A. Dalke, and K. Schulten. Vmd - visual molecular dynamics. *Journal of Molecular Graphics, 14, 33-38*, 1996.

[9] E. Krieger. Yasara - yet another scientific artificial reality application.

[10] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM SIGGRAPH Computer Graphics, 21(4)*, 1987.

[11] M. F. Sanner, A. J. Olson, and J. C. Spehner. Reduced surface: an efficient way to compute molecular surfaces. *Biopolymers, 38(3), 305-320*, 1996.

[12] LLC Schrödinger. The pymol molecular graphics system, , version 1.5.0.4 schrdinger, llc., 2010.

[13] M. Totrov and R. Abagyan. The contourbuildup algorithm to calculate the analytical molecular surface. *Journal of Structural Biology, 116, 138-143*, 1996.

	CAVER Viewer 128^3 voxels	CAVER Viewer 256^3 voxels	PyMol	VMD	YASARA
1CQW (2 754 atoms)	3,5 sec	23 sec	1 sec	3 sec	2sec
1AON (58 874 atoms)	5,5 sec	38 sec	30 sec	4-5 mins	-
70S (146 558 atoms)	6,7 sec	1 min	-	-	-

Table 1: Comparison of computational time for molecular surface in different software tools.

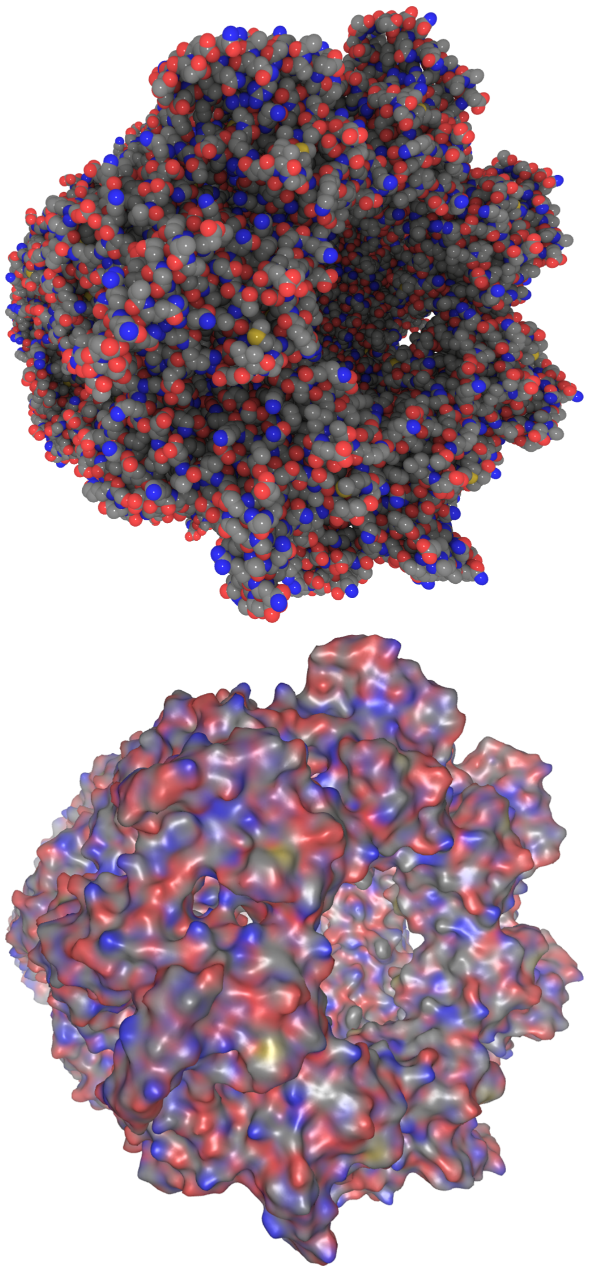
Figure 13: 1AON structure (58 874 atoms) and its molecular surface generated using a grid of 256^3 voxels.

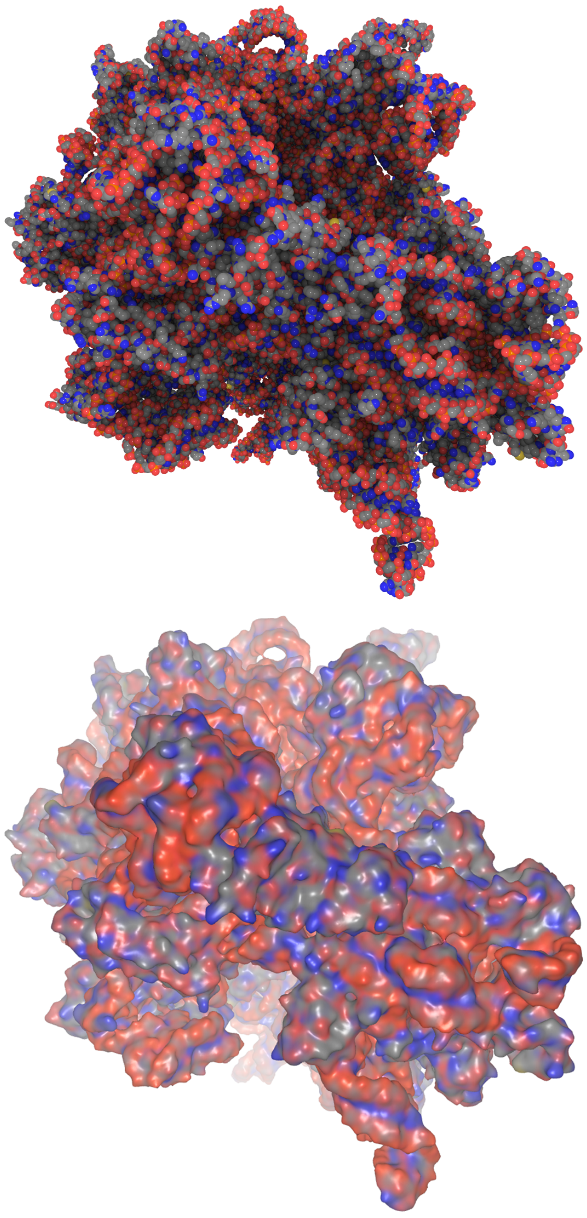
Figure 14: 70S ribosome (146 558 atoms) and its molecular surface generated using a grid of 256^3 voxels.