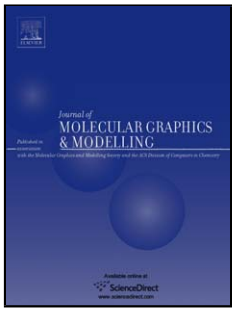


Title: MetaMol: High quality visualization of Molecular Skin Surface

Authors: Matthieu Chavent, Bruno Levy, Bernard Maigret

PII: S1093-3263(08)00053-3
DOI: doi:10.1016/j.jmgm.2008.04.007
Reference: JMG 5773

To appear in: *Journal of Molecular Graphics and Modelling*

Received date: 11-2-2008
Revised date: 17-4-2008
Accepted date: 22-4-2008

MetaMol: High quality visualization of Molecular Skin Surface

Matthieu Chavent[1*], Bruno Levy[2] and Bernard Maigret[1]

Abstract

Modeling and visualizing molecular surfaces is an important and challenging task in bioinformatics. Such surfaces play an essential role in better understanding chemical and physical properties of molecules. However, constructing and displaying molecular surfaces require complex algorithms.

In this article we introduce MetaMol, a new program that generates high-quality images in interactive time. In contrast with existing software that discretizes the surface with triangles or grids, our program is based on a GPU accelerated ray-casting algorithm that directly uses the piecewise-defined algebraic equation of the Molecular Skin Surface. As a result, both better performances and higher quality are obtained.

Keywords: Molecular Skin Surface, mixed complex, ray-casting, smooth molecular surface, GPU.

[1] Orpailleur group, Nancy University, LORIA, France.
[2] ALICE group, Nancy University, LORIA, France.
*Corresponding Author: chavent@loria.fr

1 Introduction

Visualizing molecular surfaces is important since such surfaces play a central role in molecular interactions: depicting the surfaces is essential in understanding, for example, protein-protein or protein-ligand assemblies.

Several surface definitions exist (see figure 1):

- The *Van der Waals (VdW) Surface* was firstly defined as the topological boundary of atom spheres. This surface gives a good approximation of molecular surface for small molecules but becomes rapidly complicated due to gaps and crevices. Therefore, it is not suitable for large molecular systems [1].
- To describe surfaces of macromolecules and their possible interactions with water molecules, Lee and Richards introduced the *Solvent Accessible Surface* (SAS) [2]; this surface is obtained by rolling a probe sphere depicting a water molecule over the Van der Waals surface. The trajectory of the rolling sphere center traces out the SAS.
- A few years later, Richards defined a smooth molecular surface [3] constructed as the union of two parts: the contact surface and the re-entrant surface. The contact surface is the part of the Van der Waals surface that is accessible to the probe sphere. The re-entrant surface is composed of probe surface when probe sphere rolls over pairs of atom. This surface representation suffers from self intersections. For this reason, Greer and Bush [4] decided to define the *Solvent Excluded Surface* (SES). Connolly has given an analytical method to compute this surface [5] commonly referred as the *Molecular Surface (MS)*.

- More recently, in the more general context of computational geometry, Edelsbrunner proposed to define a smoother surface: the *Skin Surface* [6]. This surface can be used to represent a molecular surface and is named *Molecular Skin Surface* (MSS). It is close to the MS representation but has extra-properties such as smoothness and decomposability. The MSS has therefore several advantages as compared to other molecular surface definitions [6]: (1) The surface does not self intersect and (2) is everywhere tangent continuous. Moreover, MSS is composed of quadrics - whereas MS comprises torus slices - which simplify calculations.

Starting from the pioneering algorithm proposed by Connolly [7], numerous works have been devoted to the improvement of methods relating to calculate and represent Molecular Surface. The goals were still the same: providing fast and robust methods that allow generating high quality pictures of MS in real time. In 1994, Varshney *et al.* developed a program that was easily parallelizable [8]. The year after, Sanner proposed a method based on reduced surfaces [9] to visualize large molecules (more than 10,000 atoms). More recently, Can *et al*. proposed an algorithm to generate Molecular Surface using level-set methods [10] and Bates *et al*. defined a *Minimal Molecular Surface* [11]. These latter two methods use a grid and marching front to display molecular surface and cavities. All these approaches are efficient but suffer from precision problems: in Varshney and Sanner algorithms, the molecular surface is triangulated while, for the Can and Bates algorithms, the surface is represented as the union of cubes so that, a level of zoom is always found where triangles or cubes appear. Some works already triangulate Skin Surface [12-14]. This has two drawbacks: (1) they need to make sure that the surface topology is preserved. This makes the algorithm complicated (and slow). (2) At a certain level of zoom, the triangles generate display artifacts (see figure 9).

To overcome these limitations, we propose to use a ray-casting method performed on Graphics Processing Unit (GPU). This has two advantages: (1) the ray-casting algorithm directly uses the equation of the MSS and does not need to resample it. (2) pixel-accurate images are generated in interactive time. GPU ray-casting has been already used to represent simple molecular models as “CPK” or “Balls and Sticks” [15, 16]. In this paper, we deal with the much more complicated case of MSS. We present the MetaMol program devoted to display in interactive time MSS with the best visualizing quality.

2 Methods

In [6], H. Edelsbrunner defines the Skin Surface as the boundary of an infinite family of spheres defined from a set of weighted points (more explanations below). The Skin Surface has the interesting property of being piecewise-defined as a set of quadratic patches. It is this latter property that is beneficial in our case: using the ray-casting approach, we can display the surface from its equation, without triangulating it nor using a grid.

In this section, we will present briefly the Molecular Skin Surface description and properties and how we can exploit them with the ray-casting approach. The reader who wishes more background material about Skin Surface is referred to [6] for the original introduction of Skin Surface and to [17, 12, 13] for detailed explanations.

2.1 Weighted Points

The inputs data of MSS are weighted points with atomic coordinates of a 3D molecular structure and a weight associated to each point. A weighted point is a pair $p = (z, w_p)$ of an atom position $z \in \mathbf{R}^3$ and a weight $w_p \in \mathbf{R}$. To make the Molecular Skin Surface wrap around atom spheres (as explained in [12]), it is necessary to define the weight as:

$$w_p = (1/s)*r_{vw}^2 \quad (1)$$

Where s is named the *shrink factor*, with $0 < s \leq 1$ and r_{vw} is the Van der Walls radius of the sphere centered on each atom. Weighted points lists or PDB files [18] can be used as inputs for MetaMol: in this case, atom radii are converted into weights using equation (1).

2.2 Molecular Skin Surface

2.2.1 Definition

Molecular Skin Surface is defined by:

- a set of weighted points (P):

$$\mathrm{P} = \{p_i = (z_i, w_i) \text{ in } \mathbf{R}^3 \times \mathrm{R} \mid i = 1..\mathrm{n}\}$$

Where p_i is a weighted point with z_i as the position of i^{th} atom (among n atoms) and w_i as its weight.

- a shrink factor s, with $0 < s \leq 1$.

The Skin Surface skn ^{s}(P) is its boundary of bdy^{s}(P) (see figure 2):

$$\mathrm{skn}^{s}(\mathsf{P}) = \partial\, \mathrm{bdy}^{s}(\mathsf{P})$$

Where bdy^{s}(P) is an infinite family of shrunk spheres generated from the finite set of weighted points (P) and named the *body* (see Section 4 of [6] for details). ∂ denotes the boundary of the union of the set of balls.

2.2.2 Equation of the MSS

Using this definition, Edelsbrunner proved that the MSS is a piecewise quadratic surface. This property is interesting for us since quadratic surfaces can be easily ray-casted. The intersection between a ray and a quadratic surface can be obtained in a closed form. The pieces of the MSS correspond to the cells of a geometric structure called the *mixed complex*.

The mixed complex, associated with a shrink factor *s*, may be thought of as an intermediate structure between the weighted Delaunay Tetrahedralization and the weighted Voronoï Diagram. The weighted Delaunay Tetrahedralization, or Regular Tetrahedralization, is a collection of points that form tetrahedra. The weighted Voronoï Diagram is the dual of the weighted Delaunay Tetrahedralization: basically, each node of Voronoï Diagram is the center of circumscribed sphere to each tetrahedron (see figure 3-A). Each mixed cell, in the mixed complex, is obtained by taking the Minkowski sum of Delaunay and Voronoï cells (see figure 3-B):

$$\mu^{s}_{X} = s \cdot \upsilon_{X} \oplus (1 - s) \cdot \delta_{X} \qquad (2)$$

Where X is a subset of a finite set of spheres and $\mu^{S}{}_{X}$ represents the mixed complex cell, υ_X the Voronoï cell and δ_X the Delaunay cell. The symbol $\cdot$ denotes the multiplication of a set by a scalar and $\oplus$ denotes the Minkowski sum. Within the mixed cell $\mu^{S}{}_{X}$, the MSS is completely determined by, at most, four weighted points in X. For *s* tends towards 0, the mixed cell tends towards the Delaunay cell. When *s* increases it deforms affinely into the Voronoï cell until *s* = 1.

The mixed complex is divided into four different parts (see figure 4). First, shrunk Voronoï cells stem from Delaunay vertices. Then, shrunk tetrahedra stem from Delaunay cells. These two cells clip pieces of sphere. Between these two objects "species", created from the Voronoï Diagram and the Delaunay Tetrahedralization, there appear two other polyhedra: prisms with shrunk Voronoï facets at their base (that we called H1 patches) and prisms with shrunk Delaunay triangles at their base (that we called H2 patches). For H1 (resp. H2) patch, the cell clips a one-sheeted or a two-sheeted hyperboloid with its symmetry axis aligned along the Delaunay (resp. Voronoï) edge.

2.3 Calculation Pipeline

In contrast with programs that create Molecular Surfaces by using the Central Processing Unit (CPU) alone for the calculations, our approach splits the calculations:

- The Mixed complex envelope is calculated on the CPU. Note that the mixed complex computation is view-independent and is therefore computed once only at the loading time.
- The ray-casting, to display MSS, is performed on GPU.

The use of CPU and GPU reduces computation times and improves the quality of the surface (see figure 5-B for a global view).

2.3.1 CPU calculations

Firstly, we construct the Delaunay Tetrahedralization using atoms centers as vertices. Then, we calculate the Voronoï Diagram from the Delaunay cells. From these two structures, we create mixed cells as a function of *s* using equation (2). Finally, we calculate the quadratic equation for each mixed complex cell using the computational geometry library CGAL [19].

2.3.2 GPU calculations

Then, we display the mixed complex envelope using the OpenGL API (figure 5-A left picture). We obtain the surface by using the ray-casting method (figure 5-A center and right pictures). In classical programs that perform ray-tracing, such as POV-Ray, each ray is traced from each pixel of the screen to the object and, when a ray intersects the object surface, a reflecting ray is calculated to define the lighting. In our case, we perform ray-casting, which is simpler and ignores secondary reflections: for each mixed complex cell, rays are launched from screen, like in the ray-tracing method, but when they touch the mixed cell, the intersection between the implicit surface and ray is calculated. If there is an intersection, the color of the pixel is computed by a simple shading model, and the Z-buffer coordinates are updated at the actual ray-surface intersection. If there is no intersection, the pixel is discarded [20]. This part is performed on the GPU, using the OpenGL Shading Language (GLSL). More precisely, the graphic card must support “Shader Model 3.0”. In practice, it is necessary to have a Geforce 6 series (respectively ATI Radeon X1300) or greater.

3 Results and discussion

The main benefit of our method is the quality and the smoothness of the surface at each level of zoom thanks to the ray-casting method. The generated images are pixel accurate. Note that thanks to the high efficiency and parallel power of the GPU, the viewpoint and the shrink factor can be changed interactively.

3.1 Molecular Skin Surface Vs Molecular Surface

To highlight our choice of surface, we compared the smoothness of the Molecular Surface (obtained with MSMS program included in VMD [21] package) with the Molecular Skin Surface. As it is presented in figure 6, the cusps observed on the MS surface are replaced, on the Skin Surface, by two-sheeted hyperboloids so that the surface is clearly continuous. Beyond the unquestionable displaying quality, the skin surface smoothness is suitable to compute and display physical and chemical properties of molecules. In contrast with the MS, where singularities are encountered [22, 23, 9, 11], the Molecular Skin Surface does not self intersect, there is no cusps and the surface is C^1 (continuous and its first order derivatives are also continuous).

Another advantage of the MSS is the ability to easily generate a smoothed SAS (Solvent Accessible Surface) by adding the probe radius to the Van der Waals radius when we define atom weight.

3.2 Deforming Molecular Skin Surface

The Molecular Skin Surface is also well adapted to support molecular deformations as this surface can be freely deformed with smooth transitions [24, 6]. On contrary to programs which compute the Molecular Skin Surface as a mesh [25, 13], using GPU ray-casting allows to display deformations in real time.

3.2.1 In changing the Shrink factor

As shown in figure 7, it is possible to interactively change the shrink factor, going from a Van der Waals surface (s = 1.0) where all atoms are represented by spheres, via the Molecular Skin Surface (s = 0.5) which is close to the MS, ending in a simplified surface if we continue to decrease the shrink factor. When the shrink factor is increased after 0.5, minor details (as reentrant regions) tend to be removed from the scene. This is interesting to have a global perception of the structure and to focus on the global shape in order to compare, for example, several protein figures [26]. Furthermore, this “simplified representation” can be useful for docking in an high-throughput approach where low-resolution models are used to perform fast calculations [27, 28] and can serve as starting points for further structural refinements [29]. When going from a representation to another, it is noticeable that the surface changes continuously.

3.2.2. During Molecular Movement

Efficient visualization of molecular movements is an important tool to understand structural behavior of biological samples. Efforts are made this last decade to reach this goal: Eyal and Halperin dynamically maintained the Van der Waals and Solvent Accessible surfaces [30]. Hao *et al.* developed a linear scalable program to visualize large time-varying molecules using occlusion culling [31]. Recently, Lampe *et al.* defined a two-level approach to visualize protein dynamics [32]. Unfortunately, these works use quite simple representation as “ball and stick” or union of spheres and few works are dedicated to represent deformations on more complicated surface such as MS [33, 34]. One explanation is that maintaining a mesh during molecular movements is very costly in computation time. In our case, with Ray-casting, there is no mesh and the precomputation time is reduced so that it is possible to visualize real-time deformations on surface. For the moment, it is possible to visualize little proteins movements in concatenated PDB file formats. But, in near future, MetaMol would be able to read Molecular Dynamics trajectory files and to display the associated moving surface in real time.

3.3 Discussion

3.3.1 Precision

Using ray-casting allows us to visualize the surface of large objects (see figure 8) with high rendering quality. Any zoom on this representation will provide a surface that is still very precise (pixel accurate), without any loss of information (see figure 8 and 9). To our knowledge, MetaMol is the first program that interactively visualizes Molecular Surface with such a quality.

3.3.2 Performances

To have a reference, we first measured the computing time of a program that computes the same type of Skin Surface but by tessellating it [25, 12]. We then compared its performances with ours (see table 1 and figure 9). We have to separate the computing time, to create the patches and generate the surfaces and the display time (measured in *frames per second* - fps). The computing times that we obtained are clearly better for a superior display quality than Kruithof's algorithm. The main reason is that the only triangles we need to generate are the faces of the mixed complex whereas Kruithof needs a very fine triangulation that approximates the MSS. In addition, Kruithof's method needs to ensure that the generated triangulation preserves the topology of the MSS, which is time consuming [25]. The use of ray-casting overcomes this limitation by calculating the position on the surface for each pixel of the screen. The efficiency of the GPU allows updating the surface "on-the-fly" when the viewpoint and/or the shrink factor are changed. It is used to visualize the surface deformations (as presented in 3.2). However, after the pre-processing step, *i.e.* once the surface is computed, the performances of Kruithof's algorithm are better than ours: this is due to multiple read-write accesses to the Z-buffer [35]. Note that this can be improved by occlusion culling techniques, which we will implement in future work. Nevertheless, to have a surface quality equivalent to ours with Kruithof's algorithm, it would be necessary to create huge number of very small triangles, so that computing and display times would increase dramatically. However, Kruithof's method provides other benefit: it is necessary to perform calculations only once for a structure (mesh is stored in an .off file) in comparison to our algorithm in which the surface is ray-casted at each frame. Unfortunately, high resolution .off files can be memory consuming: the .off file size of the 2FZS molecule (presented in figure 8) is 439 MB (~12 millions of triangles). So, decreasing the level of details could be required to visualize and manipulate large structures interactively. With Kruithof's algorithm, the surface is meshed so that it is possible to have information about neighbors of any point on the surface. Therefore, the surface can rapidly be crossed in a scanning perspective: for example, to perform physical calculations or get information about Solvent Accessible Area. In its current status our algorithm is dedicated to visualize molecular surface and is not intended to perform calculation on the surface so that our approach and Kruithof's approach are quite complementary.

3.4 User Interface

The user interface is depicted in figure 10-B. The user can easily change the visualization parameters: in particular, lighting functions (exposure, light rotation or specular effect) or cut-off plane scroll to visualize the internal structure of the molecule (see figure 10-A).

4 Conclusion

In this article, we introduce a method to accurately and interactively visualize the MSS. With our program, several types of molecular surfaces can be visualized (in changing the shrink factor): from Van der Waals surface to coarse grained surface. It is clear that, for the moment, our algorithm is less efficient for displaying huge assemblies than algorithms cited above (as MSMS or Varshney’s program). To deal with this issue, we currently work on a multi-resolution program to visualize surfaces ranging from atomic description to Cryo-Electron Microscopy level with the same precision. We will now optimize the algorithm to visualize interactively larger structures with real time surface deformations. We plan to add new effects such as ambient occlusion (already used in Tachyon [36] and Qutemol [37]) in order to improve the perception of the global shape of large molecules.

Acknowledgments

M. Chavent is supported by CNRS/Région Lorraine. B. Levy is supported by Microsoft Research (“Geometric Intelligence” Grant). Authors would like to thank L. Provot and R. Toledo for their technical supports and B. Vallet for his fruitful discussions.

References:

1. Chapman, M. and Connolly, M. L., *Molecular surfaces: calculations, uses and representations*, *International Tables for Crystallography*, 2001, pp. 539-545.
2. Lee, B. and Richards, F. M., The interpretation of protein structures: estimation of static accessibility, J Mol Biol, 1971, 55, pp. 379-400.
3. Richards, F. M., Areas, volumes, packing and protein structure., Annu Rev Biophys Bioeng, 1977, 6, pp. 151-176.
4. Greer, J. and Bush, B. L., Macromolecular shape and surface maps by solvent exclusion, Proc Natl Acad Sci U S A, 1978, 75, pp. 303-7.
5. Connolly, M. L., Analytical molecular surface calculation, Journal of Applied Crystallography, 1983, 16, pp. 548-558.
6. Edelsbrunner, H., Deformable Smooth Surface Design., Discrete & Computational Geometry, 1999, 21, pp. 87-115.
7. Connolly, M. L., molecular surface triangulation, Journal of Applied Crystallography, 1985, 18, pp. 499-505.
8. Varshney, A., Brooks, F. P. J. and Wright, W. V., Linearly Scalable Computation of Smooth Molecular, Invited submission,, IEEE Computer Graphics and Applications, 1994.
9. Sanner, M. F., Olson, A. J. and Spehner, J. C., Reduced surface: an efficient way to compute molecular surfaces., Biopolymers, 1996, 38, pp. 305-320.
10. Can, T., Chen, C.-I. and Wang, Y.-F., Efficient molecular surface generation using level-set methods., J Mol Graph Model, 2006, 25, pp. 442-454.

11. Bates, P. W., Wei, G. W. and Zhao, S., Minimal molecular surfaces and their applications, J Comput Chem, 2007.
12. Kruithof, N. and Vegter, G., Approximation by skin surfaces, Computer-Aided Design, 2004, 36, pp. 1075-1088.
13. Cheng, H.-L. and Shi, X., *Guaranteed Quality Triangulation of Molecular Skin Surfaces, Proceedings of the conference on Visualization '04*, IEEE Computer Society, 2004, pp. 481-488.
14. Cheng, H.-L. and Shi, X., Quality Mesh Generation for Molecular Skin Surfaces Using Restricted Union of Balls, vis, 2005, 00, pp. 51-57.
15. Toledo, R. and Levy, B., Extending the graphic pipeline with new GPU-accelerated primitives, Tech report, 2004.
16. Sigg, C., Weyrich, T., Botsch, M. and Gross, M., GPU-Based Ray-Casting of Quadratic Surfaces, Symposium on Point-Based Graphics, 2006, pp. 56-65.
17. Kruithof, N. G. H., *Envelope Surfaces, Surface Design and Meshing*, University of Groningen, 2006, pp. 125.
18. Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N. and Bourne, P. E., The Protein Data Bank (http://www.rcsb.org/pdb/home/home.do), Nucleic Acids Res, 2000, 28, pp. 235-42.
19. *Cgal, Computational Geometry Algorithms Library, http://www.cgal.org*, 1999.
20. Loop, C. and Blinn, J., Real-time GPU rendering of piecewise algebraic surfaces, ACM Trans. Graph., 2006, 25, pp. 664-670.

21. Humphrey, W., Dalke, A. and Schulten, K., VMD: visual molecular dynamics., J Mol Graph, 1996, 14, pp. 33-8, 27-8.

22. Vorobjev, Y. N. and Hermans, J., SIMS: computation of a smooth invariant molecular surface, Biophys J, 1997, 73, pp. 722-32.

23. Geng, W., Yu, S. and Wei, G., Treatment of charge singularities in implicit solvent models, J Chem Phys, 2007, 127, pp. 114106.

24. Edelsbrunner, H., *Smooth Surfaces for Multi-Scale Shape Representation, Proceedings of the 15th Conference on Foundations of Software Technology and Theoretical Computer Science*, Springer-Verlag, 1995, pp. 391-412.

25. Kruithof, N. and Vegter, G., Meshing skin surfaces with certified topology, Comput. Geom. Theory Appl., 2007, 36, pp. 166-182.

26. Cipriano, G. and Gleicher, M., Molecular surface abstraction, IEEE Trans Vis Comput Graph, 2007, 13, pp. 1608-15.

27. Tovchigrechko, A., Wells, C. A. and Vakser, I. A., Docking of protein models, Protein Sci, 2002, 11, pp. 1888-96.

28. Ritchie, D. W. and Kemp, G. J. L., Fast computation, rotation, and comparison of low resolution spherical harmonic molecular surfaces, Journal of Computational Chemistry, 1999, 20, pp. 383-395.

29. Gray, J. J., Moughon, S., Wang, C., Schueler-Furman, O., Kuhlman, B., Rohl, C. A. and Baker, D., Protein-protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations, J Mol Biol, 2003, 331, pp. 281-99.

30. Eyal, E. and Halperin, D., *Dynamic maintenance of molecular surfaces under conformational changes, Proceedings of the twenty-first*

annual symposium on Computational geometry, ACM, Pisa, Italy, 2005, pp. 45-54.

31. Hao, X. V., A., *Real-Time Visualization of Large Time-Varying Molecules, Proceedings of the High-Performance Computing Symposium '04*, 2004.
32. Lampe, O. D. V., I.; Reuter, N. & Hauser, H., Two-Level Approach to Efficient Visualization of Protein Dynamics, Visualization and Computer Graphics, IEEE Transactions on, Nov.-Dec. 2007, 13, pp. 1616-1623.
33. Sanner, M. F. and Olson, A. J., Real time surface reconstruction for moving molecular fragments, Pac Symp Biocomput, 1997, pp. 385-96.
34. Bajaj, C. L., Pascucci, V., Shamir, A., Holt, R. J. and Netravali, A. N., Dynamic maintenance and visualization of molecular surfaces, Discrete Appl. Math., 2003, 127, pp. 23-51.
35. Foley, J., Van Dam, A., Feiner, S. and Hughes, J., *Computer Graphics: Principles and Practice, second edition in C*, Addison-Wesley Professional, 1995.
36. Stone, J., *An efficient Library for Parallele Ray Tracing and Animation*, *Computer Science Department*, University of Missouri-Rolla, 1998.
37. Tarini, M., Cignoni, P. and Montani, C., Ambient Occlusion and Edge Cueing for Enhancing Real Time Molecular Visualization, IEEE Transactions on Visualization and Computer Graphics, 2006, 12, pp. 1237-1244.
38. Krieger, E., *YASARA (www.yasara.org)*, 2003.

Figures Caption:

Figure 1:

Surface definitions. Van der Waals and Molecular surfaces were created using VMD [21]. Solvent Accessible Surface was created using YASARA [38]. These three images were post-processed using POV-Ray. Molecular Skin Surface was obtained using MetaMol.

Figure 2:

The Skin Surface in 2D of two weighted points (in dashed red). The green disks form a subset of the *body* whose boundary (in black) is the skin curve.

Figure 3:

A- The Voronoï Diagram (dashed line) and its dual: the Delaunay Triangulation (solid line) in 2D. B- The mixed Complex (for *s* = 0.5) based on Delaunay Triangulation and Voronoï Diagram: the Voronoï and Delaunay cells are shrunk and we can notice the appearance of new patches between cells.

Figure 4:

From Tetrahedralization elements, it is possible to obtain mixed complex cells. These cells clip one of the 3 objects which constitute the Skin Surface: the sphere, the one-sheeted hyperboloid and the two-sheeted hyperboloid.

Figure 5:

A- Visualization of the C_{60} fullerene molecule. On the left, the primitive envelope of patches. On the right, the ray-casted result. At the center the combined ray-casted and primitive representations. *Colors on the surface correspond to the different mixed cells: green = shrunk Voronoï cells, red = shrunk tetrahedra, yellow = H1 patches and pink = H2 patches.* B- Pipeline of our algorithm.

Figure 6:

Comparison of Molecular Skin Surface (top) and Molecular Surface obtained with MSMS program included in VMD package (bottom) with the maximum density of points. *S* represents the shrink factor and r_p represents the radius of the rolling probe sphere.

Figure 7:

Representation of the molecule 200d. In function of the shrink factor we pass from a near Van der Waals representation (for s = 0.9) to the near MS representation (for s = 0.5) to a simplified form (for s = 0.1). *Colors on the surface correspond to the different mixed cells: green = shrunk Voronoï cells, red = shrunk tetrahedra, yellow = H1 patches and pink = H2 patches.*

Figure 8:

Visualization of the E. coli ClpP (pdb ID: 2FZS) containing 20620 atoms. Zoom on this macromolecule shows a very smooth surface without loss of information.

Figure 9:

Visualizing the Gramicidin A molecule, with Kruithof's approach (with and without wire frame) and with our method.

Figure 10:

A-Visualizing the interior of molecule 2FZS thanks to a cut plane. *Cutplane representation in transparent and dashed line was added to emphasize the image*. B- Viewer panel.

Table 1:

Performance of our approach compared with Kruithof's approach. We use an Intel CPU at 2.40 GHz with a Nvidia GeForce 8800 GTX graphic card. Note that hetero atoms in PDB files were removed before measurement.

PDB code	No. atoms	Kruithof 's approach			MetaMol		
		Nb of triangles	Computing time (s.)	FPS (1024x1024)	Nb of triangles	Computing time (s.)	FPS (1024x1024)
7tmn	33	23,424	1.1	800	7,116	0.02	200
1grm (*Gramicidin A*)	272	310,488	16.1	130	73,416	1.7	50
1g6x	509	481,856	28.7	95	146,476	3.6	25
1cbs	1091	1,664,184	93.1	30	325,076	8.2	12
1j4n	1852	2,165,268	137.4	25	558,372	15.4	7

Figure1

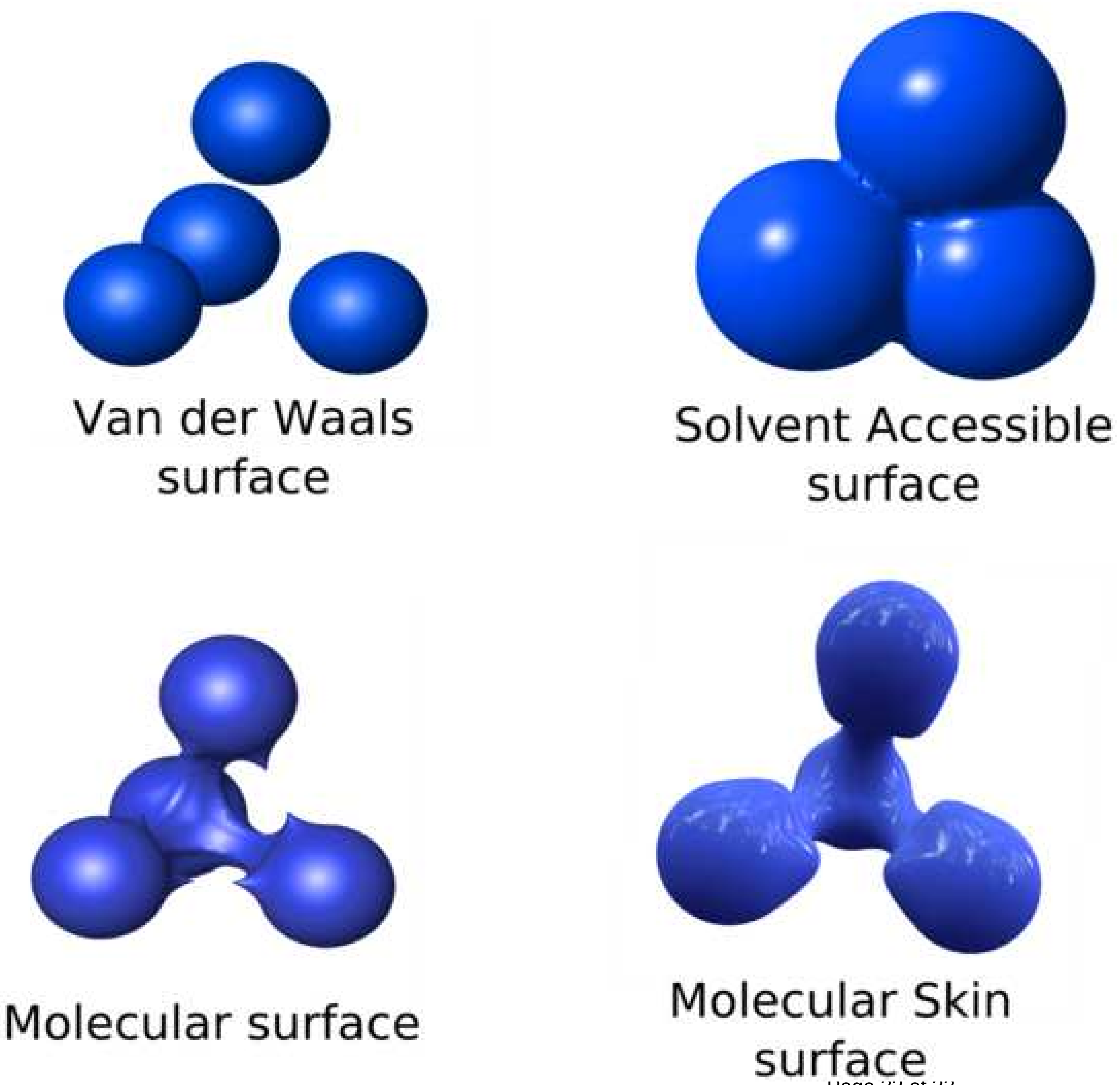

Van der Waals surface
Solvent Accessible surface
Molecular surface
Molecular Skin surface

Figure2

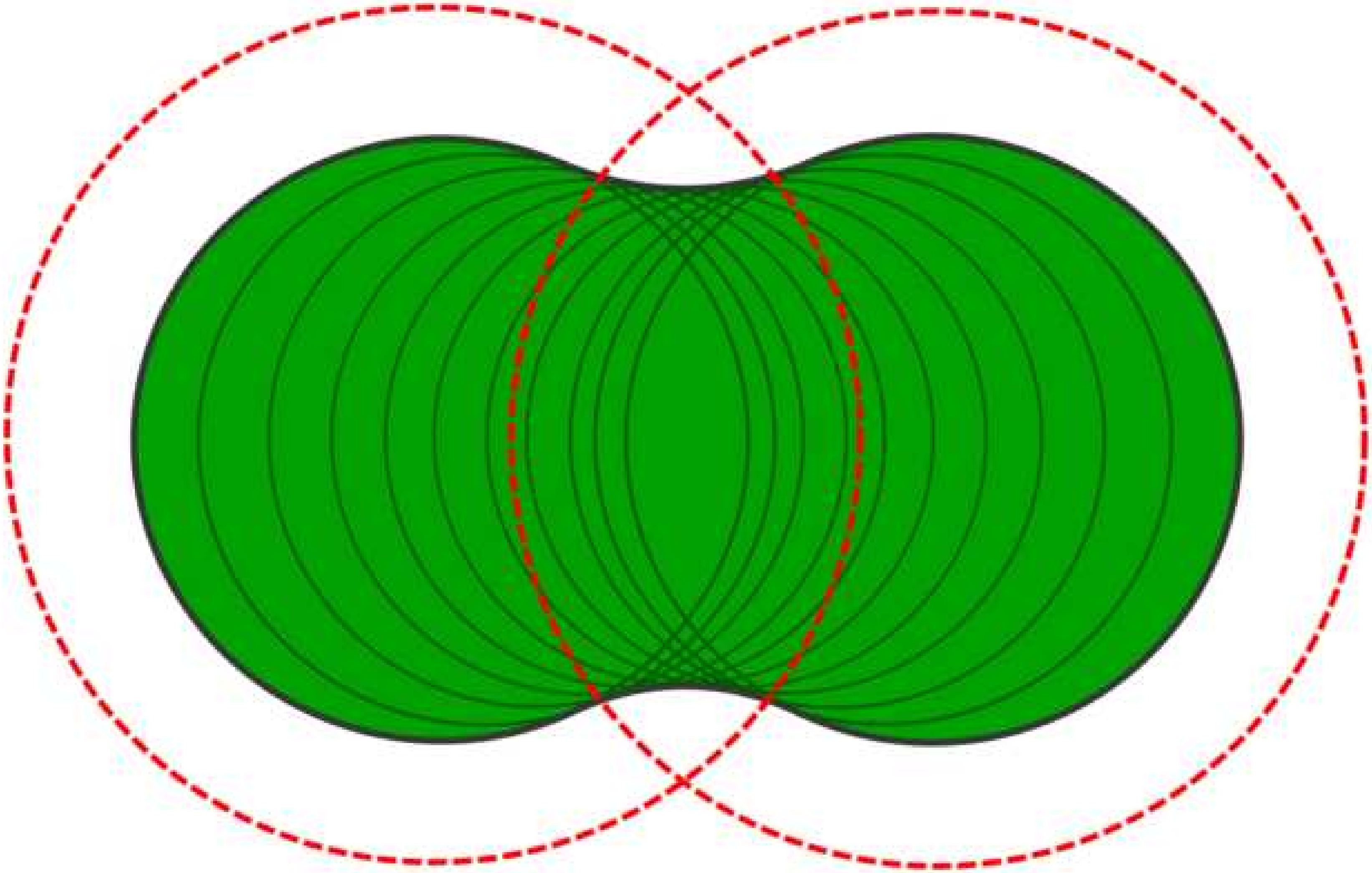


Figure3

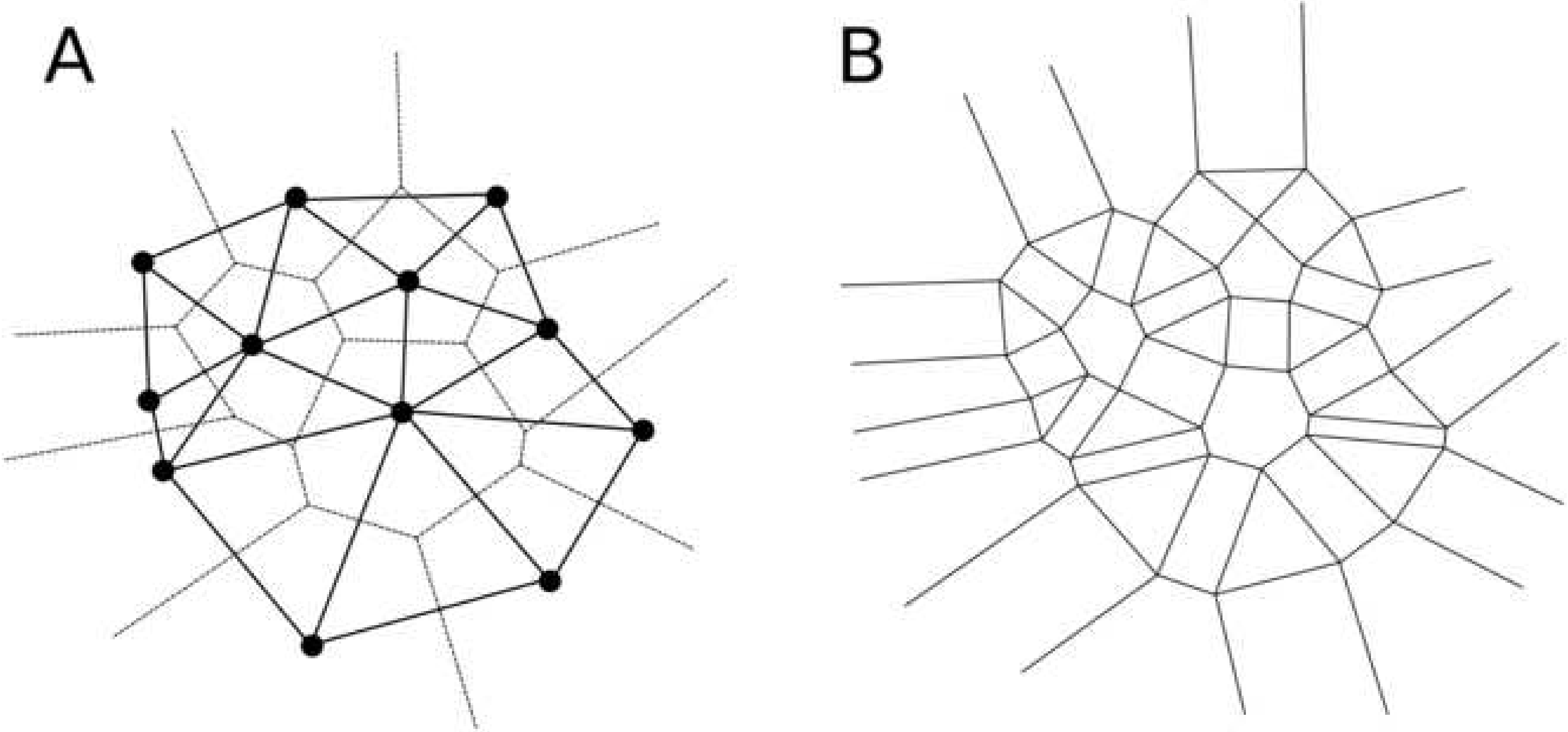


Figure4

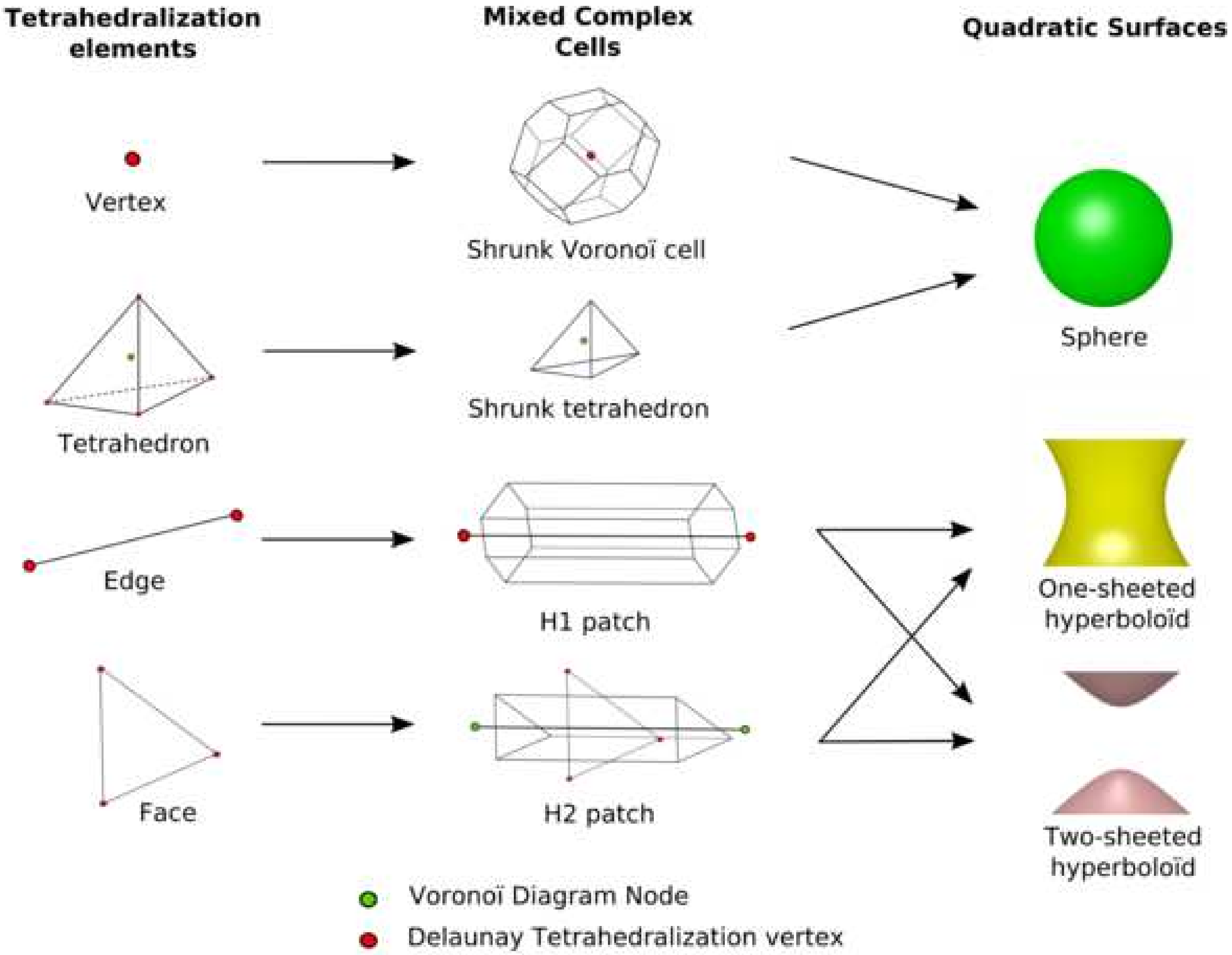
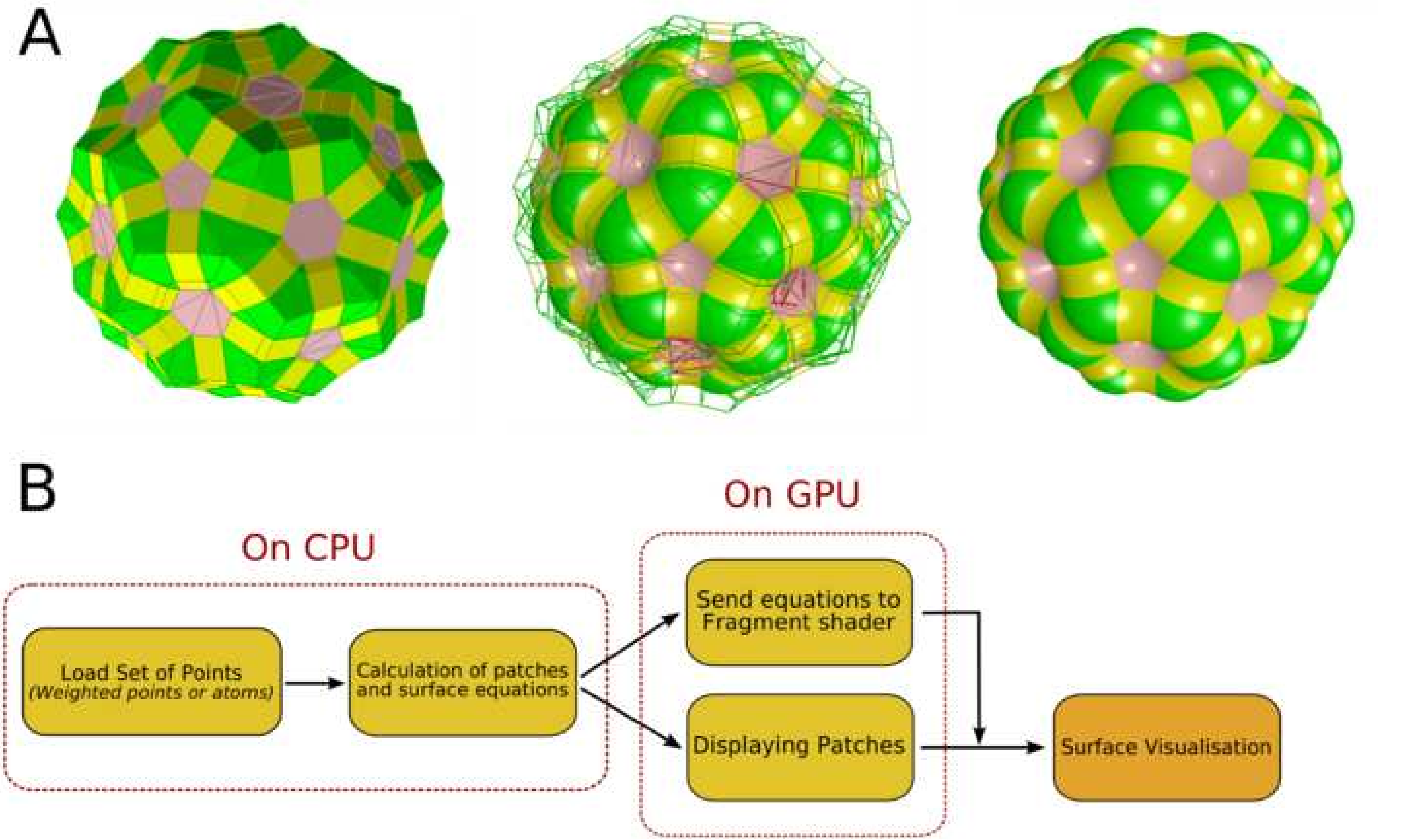
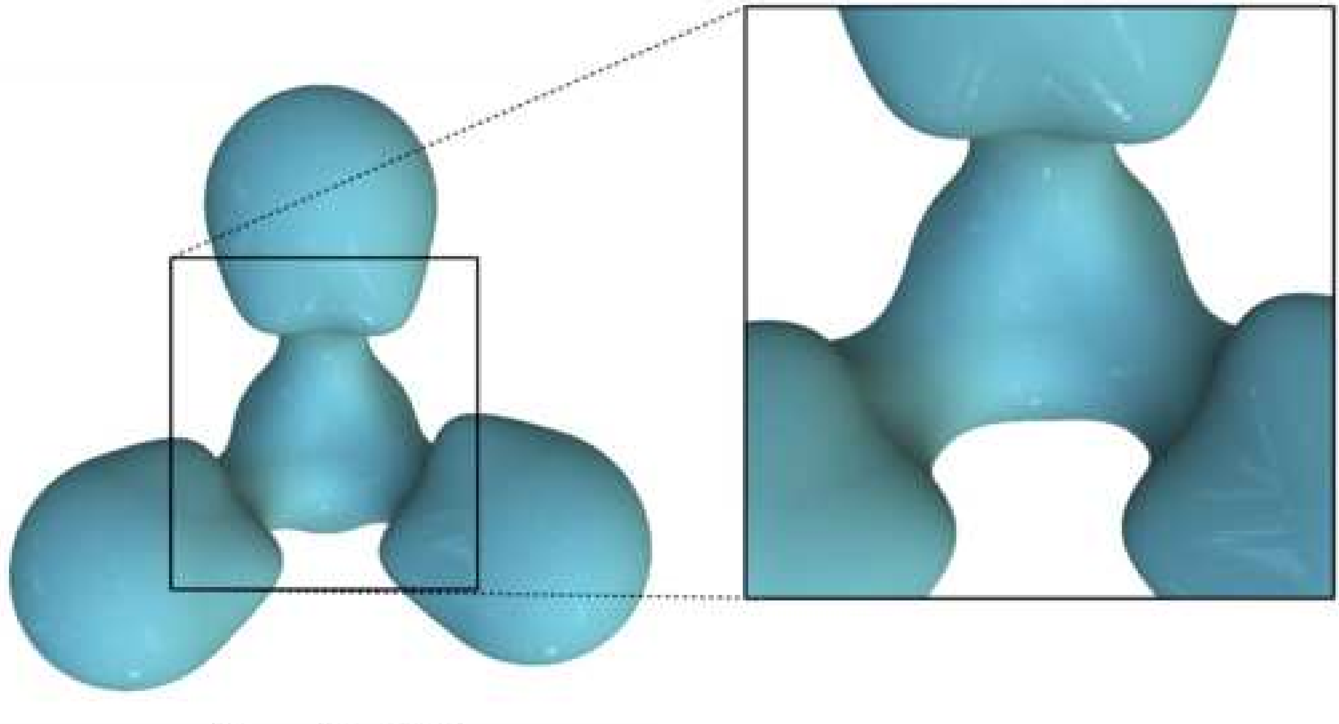
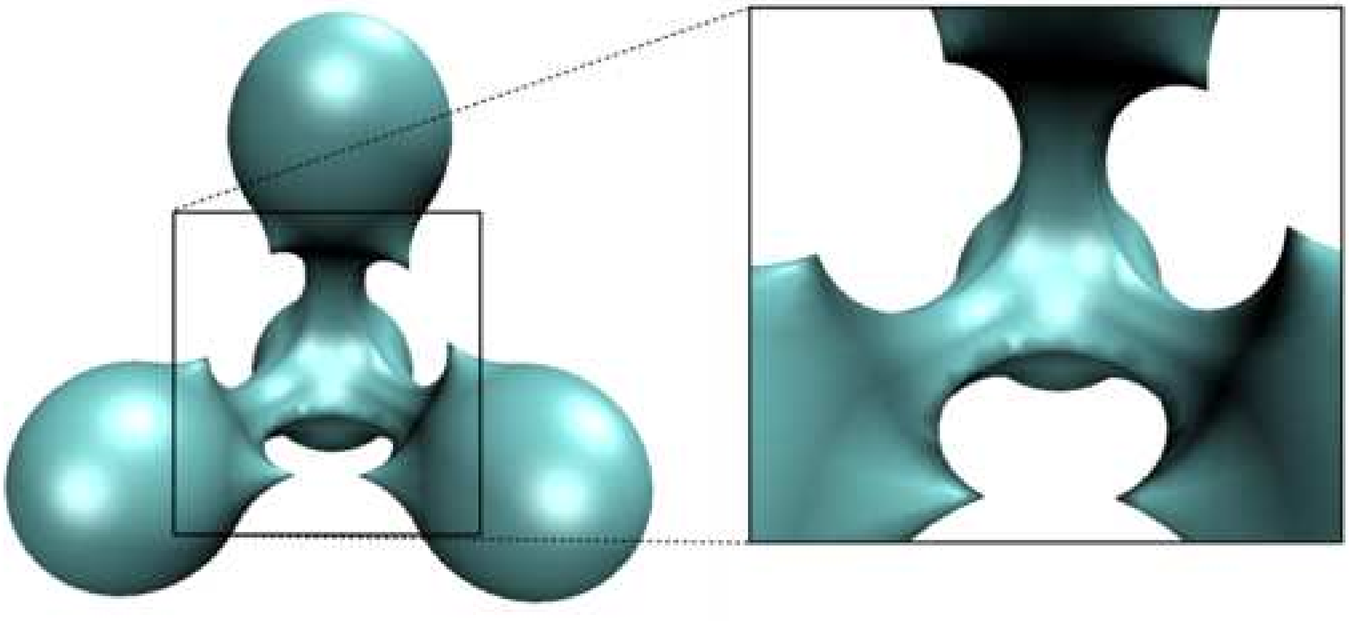

Tetrahedralization elements
Mixed Complex Cells
Quadratic Surfaces
Vertex
Shrunk Voronoï cell
Sphere
Tetrahedron
Shrunk tetrahedron
Edge
H1 patch
One-sheeted hyperboloïd
Face
H2 patch
Two-sheeted hyperboloïd
Voronoï Diagram Node
Delaunay Tetrahedralization vertex

Figure5


A
B
On GPU
On CPU
Load Set of Points
(Weighted points or atoms)
Calculation of patches and surface equations
Send equations to Fragment shader
Displaying Patches
Surface Visualisation

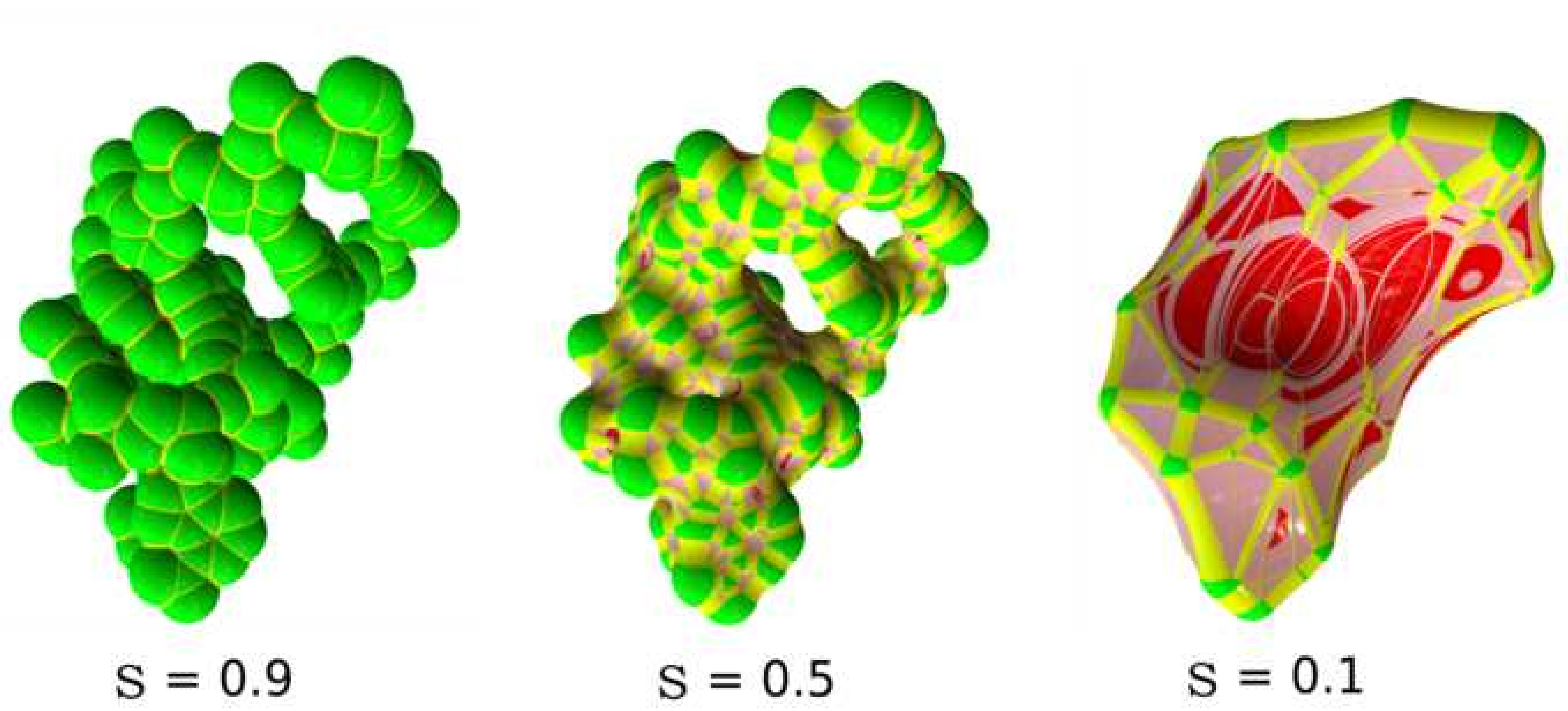


S=0.40



r_p =1.4

Figure7



s = 0.9

s = 0.5

s = 0.1

Figure8

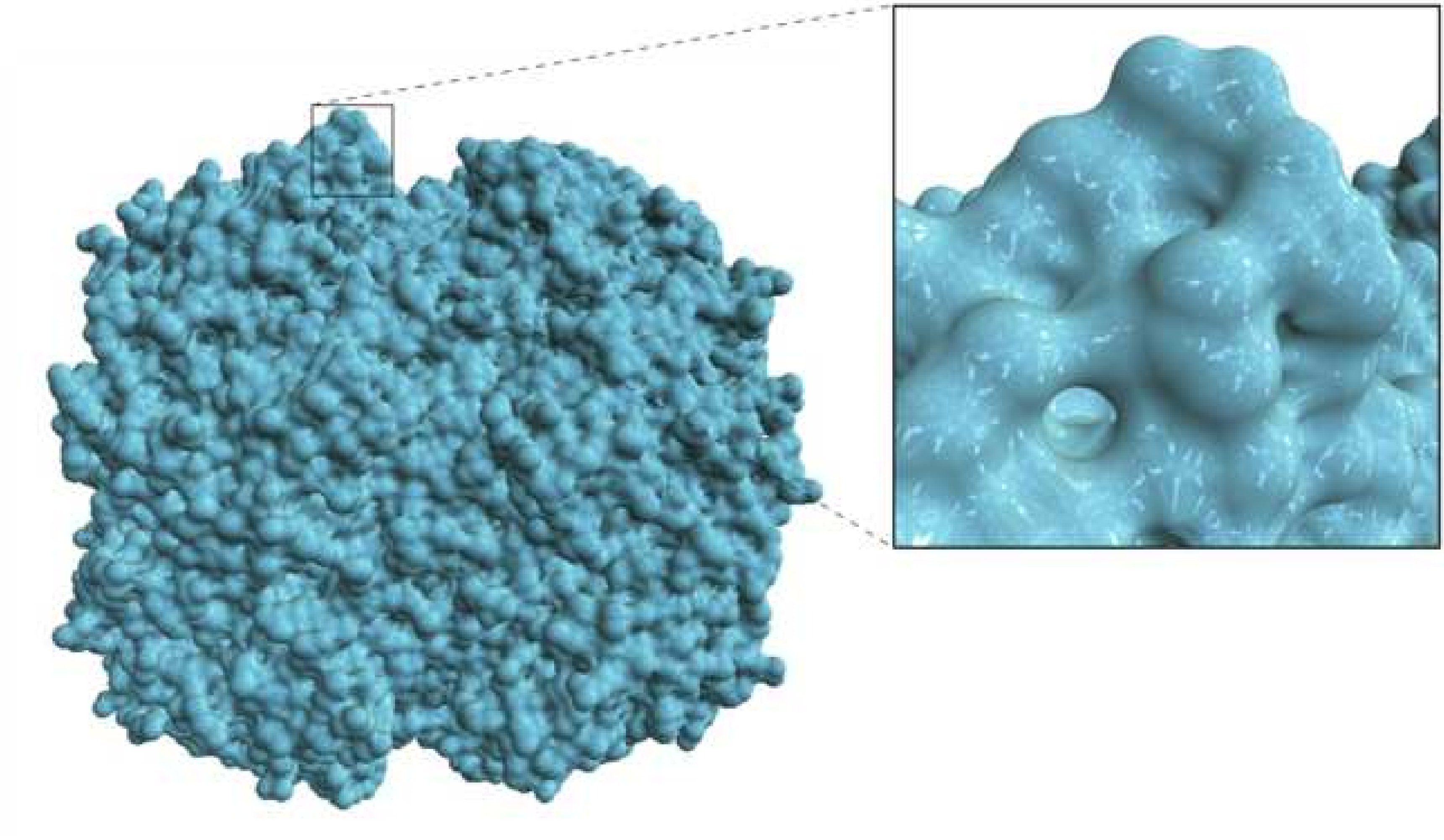


Figure9

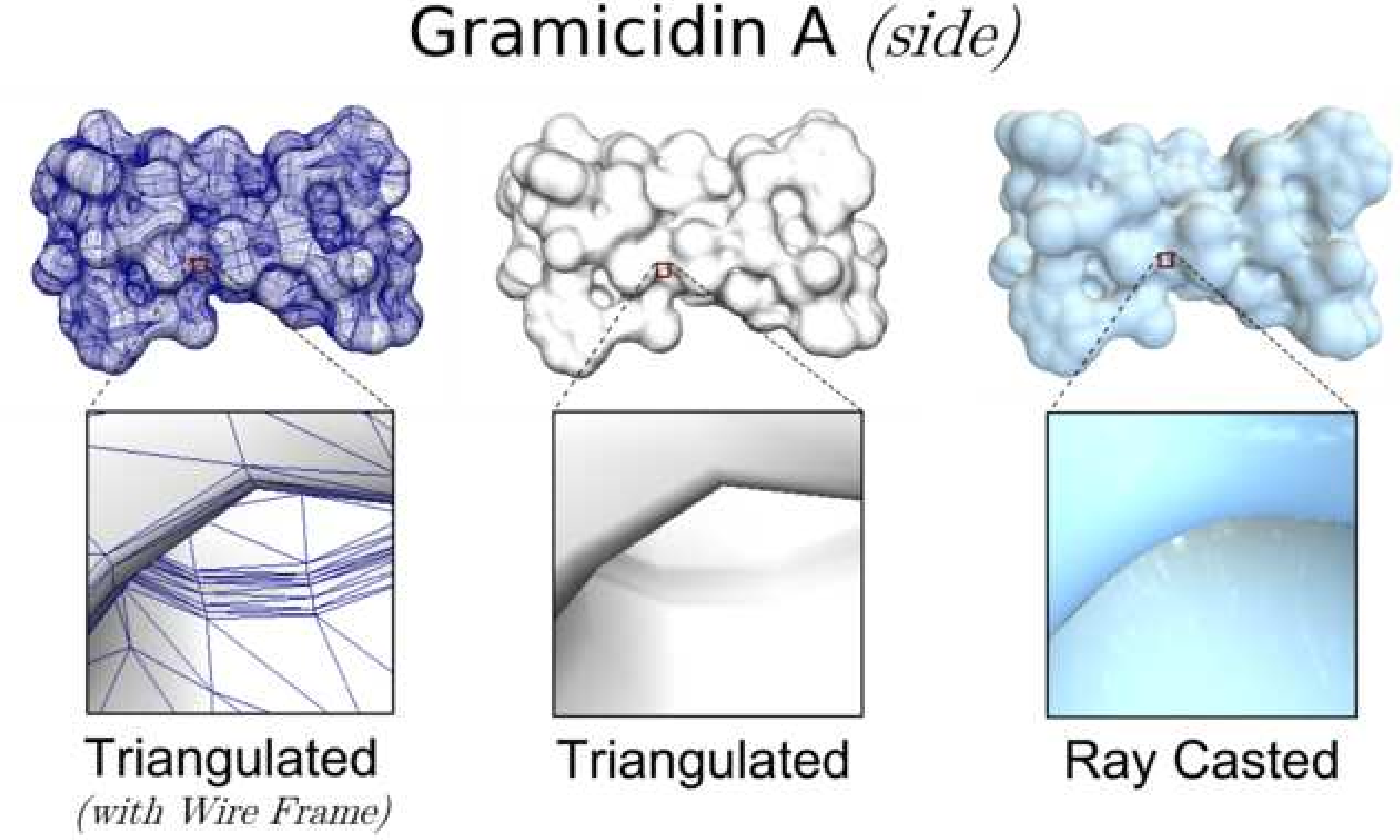
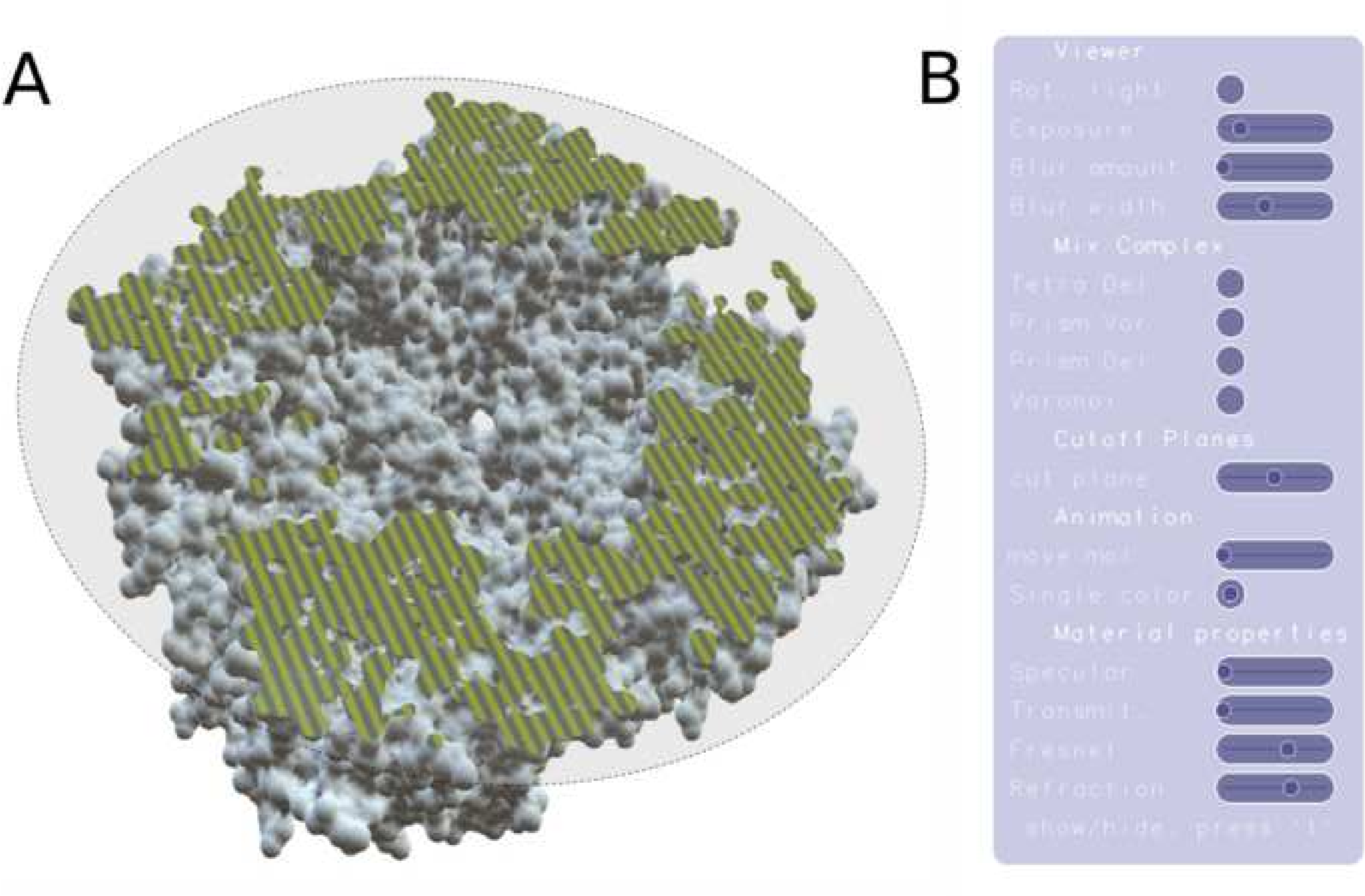

Gramicidin A *(side)*
Triangulated
(with Wire Frame)
Triangulated
Ray Casted

Figure10


A
B
Viewer
Rot. light
Exposure
Blur amount
Blur width
Mix Complex
Tetra Del
Prism Vor
Prism Del
Voronoi
Cutoff Planes
cut plane
Animation
move mol
Single color
Material properties
Specular
Transmit
Fresnel
Refraction
show/hide, press 'l'